

P
C/D

INFORM@TIQUE



Par:

NDOKANG ESONE Ludwick 1st
YOUMBINGONGANG Charlie
ZEFOUET Marie Chantal
MBIENDA Judith

www.dite-ens.cm

P
C/D

INFORM@TIQUE

Par:

NDOKANG ESONE Ludwick 1st

YOUMBI NGONGANG Charlie

ZEFOUET Marie Chantal

MBIENDA Judith

Promotion Arc-en-Ciel

Promotion Arc-en-Ciel

www.dite-ens.cm

PRÉSENTATION DU LIVRE

Ce manuel scolaire que nous avons élaboré et que nous espérons être compréhensibles, pratique et claire pour les apprenants est aussi un manuel de recueil pour les enseignants. Il épouse étroitement le programme officiel d'informatique de l'enseignement secondaire de septembre 2011. Il prend en compte la grande évolutivité de l'informatique en termes de matériels et de logiciels liés à l'enseignement de cette nouvelle discipline et des disciplines traditionnelles. C'est pourquoi ce manuel met plus l'accent sur les invariants matériels et logiciels (c'est-à-dire les connaissances qui ne changent pas d'un environnement de travail à un autre), et essaient ainsi de lui donner une image scientifique au-delà de son aspect artistique.

Ce manuel concerne les classes de première C et D de l'enseignement secondaire général et son essence principale est d'initier les apprenants à la réalisation de leurs propres sites web. Bien que sa première cible soit les élèves de Première C et D, il s'adresse à un public plus élargi allant de l'analphabète numérique aux initiés dans le domaine. Autrement dit et de manière plus spécifique, toute personne sachant lire et écrire le français, ayant un minimum de connaissances dans l'utilisation d'un ordinateur et de la motivation à apprendre est bien placé pour exploiter ce manuel et créer des sites web.

Pour atteindre cet objectif, les contenus de ce manuel sont organisés autour de trois principaux concepts, lesquels constituent les modules du livre. Le premier module intitulé **INFOGRAPHIE** prépare l'apprenant à la réalisation des images ou animations qu'il utilisera lors de la réalisation de son site. Le deuxième module nommé **ALGORITHMIQUE** introduit les notions d'algorithme afin de préparer l'apprenant à la programmation et particulièrement à la traduction en langage algorithmique de certains problèmes de la vie courante. Le troisième module intitulé **CRÉATION D'UN SITE WEB** initie l'apprenant à la programmation des ses pages web et donc à la réalisation de son propre site web. Les trois modules trouvent leur convergence dans un **PROJET** de réalisation de site web statique où l'apprenant est appelé à mettre en œuvre toutes les compétences acquises tout au long de sa formation. Nous avons complété le programme en y insérant un chapitre introductif aux notions qui seront vues par les apprenants en classe de Terminale. Ce chapitre essaie également de dépasser le cadre restreint auquel est sujet le programme en prenant en compte l'évolution que connaît le monde informatique, et plus particulièrement le domaine de création des sites web.

Chaque module est subdivisé en chapitres. Chaque chapitre commence par une page préparatoire qui présente le titre du chapitre, l'illustration liée au thème et le plan du chapitre. La deuxième page du chapitre présente la situation problème du chapitre, les pré-requis, les objectifs et l'introduction au chapitre. Après, suivent des pages subséquentes

qui présentent le contenu du chapitre. La fin de chaque chapitre se termine par une évaluation des connaissances afin de vérifier le niveau d'acquisition de la leçon. L'activité d'intégration mobilise les savoirs et savoir-faire des élèves pour résoudre les cas pratiques.

Les différents chapitres ont une moyenne de dix pages et comportent les différents éléments expliqués ci-après :

 Un Numéro :

Précédé du mot « *chapitre* », il permet d'identifier la position du chapitre dans le livre.

 Un Titre

Il donne une idée générale du contenu. Celui-ci est court, précis et parlant afin de ne pas prêter à confusion.

 Une Image illustrative

Elle permet à l'apprenant de se faire une orientation sur le sujet qui sera traité dans ledit chapitre. Celle-ci est soit libre, soit appartenant aux auteurs du livre.

 Un Sommaire

Il présente les grands axes du chapitre.

 Une Situation problème

Elle suscite l'intérêt de l'apprenant en posant un problème qui pourra être résolu tout au long du chapitre ou par l'apprenant à la fin du chapitre.

 Les Pré-requis :

Ce sont les connaissances et les compétences de base que doit avoir l'apprenant pour une bonne compréhension du chapitre. Elles regroupent les savoirs, les savoirs faire et savoirs être afin de permettre une acquisition aisée des connaissances et des compétences nouvelles.

 Les Objectifs :

Ce sont des connaissances et compétences que l'apprenant doit acquérir à la fin du chapitre. Elles sont regroupées en savoirs et savoirs-faires afin de faciliter l'organisation des contenus pédagogiques.

 Une Introduction :

Elle introduit les notions qui seront développées dans le chapitre tout en intégrant soit un petit historique du concept, soit l'intérêt à étudier la notion.

 Les Contenus simples :

C'est le cours proprement dit. Ils détaillent les éléments annoncés dans le sommaire.

 Les Remarques :

Comme son nom l'indique, elles mettent en relief un ou plusieurs concepts auxquels l'apprenant devrait prêter attention.

 Les Activités :

Cette partie présente des travaux pratiques et exercices permettant à l'apprenant de mettre en application les notions acquises dans une partie de la leçon. Ces activités sont

faites en petits groupes d'apprenants afin de susciter l'entraide entre eux et un nivellement des connaissances et des compétences.

Les Applications

Elles permettent de faire une évaluation intermédiaire de l'apprenant afin de s'assurer que celui-ci a atteint le niveau requis. Elles donnent l'occasion à l'apprenant de s'exercer, d'assimiler un savoir et d'auto-développer les compétences attendues.

Les Compléments :

Ils interviennent par endroit et ont pour rôle principal d'apporter des suppléments de connaissances.

Les Astuces :

Elles donnent des techniques d'apprentissage, des méthodes de résolution des problèmes. Bref, elles donnent des raccourcis à l'apprenant.

Un Résumé :

Celui-ci mentionne les points essentiels et les idées générales de la leçon et se place à la fin du chapitre. Ce rappel facilite la rétention de ces points et de ces idées et les fixe dans l'esprit du lecteur.

Des Exercices :

Cette partie renferme un ensemble d'exercices permettant à l'apprenant de réviser toutes les notions apprises dans le chapitre.

Les Images:

Les images contenues dans le cours permettront au lecteur de suivre, pas seulement textuellement mais aussi visuellement l'évolution des cours. Ceci facilitera la compréhension et l'assimilation des contenus présentés. Les images sont explicites et sont constituées en majorité des captures d'écran.

Les Pictogrammes :

Ce sont des éléments indicatifs du contenu qui permettent à l'apprenant de reconnaître facilement le type d'information contenu dans la partie concernée. Ils sont représentés sous forme d'icônes simples ou d'icônes accompagnés d'un texte indicatif afin de permettre à l'apprenant de facilement identifier le contenu.

TABLE DES MATIÈRES

Module 1 : INFOGRAPHIE.....	6
Chapitre 1 : Notion d'infographie.....	7
1. Présentation générale	
2. Les éléments d'infographie	
3. Utilisation d'un logiciel d'infographie	
Chapitre 2 : Création des textes avec effets.....	19
1. Présentation des outils de texte	
2. Déformation d'un texte	
3. Améliorer la production des documents de PAO	
Chapitre 3 : Retouche d'une photo et création des boutons.....	28
1. Retouche d'une image	
2. Création des boutons et logos	
Chapitre 4 : Création des animations.....	37
1. Présentation de la palette d'animation	
2. Opérations faites dans la palette d'animation	
Module 2 : ALGORITHMIQUE.....	46
Chapitre 5 : Notion d'algorithmique.....	47
1. Étapes de résolution d'un algorithme	
2. Les éléments de la structure d'un algorithme	
3. Structure générale d'un algorithme	
Chapitre 6 : Instructions simples, procédures et tableaux.....	59
1. Opérations sur les variables	
2. Les tableaux	
3. Procédures et fonctions	
Chapitre 7 : Les structures algorithmiques.....	69
1. Les structures séquentielles	
2. structures alternatives	
3. Les structures de choix	
4. Les structures itératives ou répétitives	
Chapitre 8 : Exécution d'un algorithme.....	79
1. État des variables au cours de l'exécution de quelques algorithmes	
2. Décompte du nombre d'instructions et construction d'un organigramme	

Module 3 : CREATION D'UN SITE WEB..... 88

Chapitre 9 : Notion de page web avec le HTML.....89

1. Préparation d'un environnement de programmation web
2. Notion de base du langage HTML
3. Les balises

Chapitre 10 : Insertion des liens, image et sons dans une page web.....103

1. Insertion des liens
2. Insertion des images
3. Insertion de son audio

Chapitre 11 : Insertion des tableaux et des frames dans une page web..... 115

1. Les tableaux
2. Les frames

Chapitre 12 : Insertion des formulaires et des boutons dans une page web..... 128

1. Insertion des formulaires
2. Insertion des boutons

Chapitre 13 : Production d'une feuille de style..... 139

- I. Comment appliquer une feuille de style ?
- II. Quelques propriétés de formatage du texte
- III. Modification de l'arrière plan de la page
- IV. Ou placer le code "CSS" ?

Chapitre 14 : Introduction au site web dynamique..... 152

1. Les limites du site web statique
2. Du site web statique au site web dynamique
3. Fonctionnement d'un site web dynamique

PROJET : RÉALISER UN SITE WEB STATIQUE..... 158

1. Analyse
2. Conception
3. Réalisation
4. Quelques thèmes de projets

MODULE

1

INFOGRAPHIE

- 1- Notion d'infographie*
- 2- Création des textes avec effets*
- 3- Retouche d'une photo et création des boutons*
- 4- Création des animations*

Chapitre 1

NOTION D'INFOGRAPHIE



SOMMAIRE



- I- PRÉSENTATION GÉNÉRALE
- II- LES ÉLÉMENTS D'INFOGRAPHIE
- III- UTILISATION D'UN LOGICIEL D'INFOGRAPHIE



Problème

On désire créer une école primaire dans votre village. Le chef du village lance un concours primé pour la conception d'une plaque contenant des informations sur ladite école.

1. Quelles sont les informations qu'on peut retrouver sur une telle plaque ?
2. Faites une esquisse de cette plaque sur une feuille de papier.



Objectifs

- Connaître la genèse du terme infographie
- Reconnaître les produits de l'infographie
- Concevoir des objets (textes, images, etc.)
- Définir et comprendre l'utilité de l'infographie dans la société.
- Être capable de télécharger, installer et ouvrir un logiciel d'infographie.
- Connaître les notions de base en infographie

INTRODUCTION

L'histoire de l'infographie ne date pas d'aujourd'hui comme plusieurs semblent le penser. Elle trouve ses origines dans des domaines comme la calligraphie, l'écriture, le graphisme, le dessin. C'est un art qui s'appuie sur des représentations pour passer une information.

Pourquoi un art ? Elle fait appel aux qualités esthétiques, à l'intuition et à la créativité de l'infographe.

Tout comme le graphiste a à sa disposition un ensemble d'**outils** pour faire son travail (les crayons, les couleurs, les pinceaux, la gomme, l'encre, la règle, etc.), les logiciels d'infographie mettent également à la disposition des utilisateurs les mêmes **outils** en version numérique et plus améliorés. Ces logiciels offrent de plus grandes possibilités c'est-à-dire qu'ils permettent de représenter certains aspects impossible de réaliser par simple graphisme.

Les premières représentations par ordinateur ont été produites au début des années **1950** au **Massachusetts Institute of Technology**. Au fil du temps, les avancées spectaculaires en informatique vont conduire à l'élaboration des logiciels d'infographie encore plus performants. Ces logiciels offrent des potentialités de manipulations des textes, des images et des sons qui n'étaient jusque là réservées qu'aux professionnels. Tout se passe sur l'écran d'un moniteur et le résultat des retouches est visible (presque) immédiatement grâce à des logiciels spécialisés, de plus en plus nombreux.

I. PRÉSENTATION GÉNÉRALE

I. Définition

Terme formé en 1974, l'**infographie** naît à partir des mots '**informatique**' et '**graphisme**', les deux termes sont souvent confondus. Bien que les deux soient liés, la différence se situe au niveau des techniques employées.

Le **graphisme** désigne l'activité qui consiste à créer, choisir et utiliser des éléments graphiques (dessins, caractères typographiques, images, couleurs, etc.) pour élaborer un objet de communication ou de culture.

L'**infographie** quant à elle, est définie comme étant le graphisme combiné à l'utilisation d'un ordinateur et des logiciels prévus à cet effet. L'**infographie** est également définie comme l'utilisation de représentations (texte, dessins, images) pour représenter des données ou des résultats provenant de traitements informatiques.



Infographie = Information + Informatique + Graphisme = L'Information par l'image.

L'**information** est une séquence d'évènements. On a plusieurs représentations de l'information :

- Le **texte** : c'est un ordre d'apparition des lettres de l'alphabet.
- Le **son** : c'est une succession d'ondes sonores.
- L'**image** : c'est un voisinage ou ensemble géométrique des pixels.
- L'**animation** : c'est un voisinage géométrique et temporel de pixels i.e. un ensemble de pixels s'étendant dans le temps.

L'**infographie** est donc une branche de l'informatique qui permet de transmettre une information par une représentation (texte, image, animation) et dont l'information dépend de modifications qui sont faites sur ladite représentation à partir d'un ordinateur.

L'utilisateur des logiciels d'infographie s'appelle un **infographe** ou un **infographiste**.

2. Du graphisme à l'infographie

Le passage du graphisme à l'infographie repose essentiellement sur l'utilisation de l'outil informatique. C'est la différence majeure entre ces deux domaines. On peut donc assimiler le graphiste à un infographe à la différence que le premier n'intègre pas la composante informatique dans son œuvre artistique pour certains et scientifique pour d'autres.

On rencontre dans la vie quotidienne des exemples concrets de graphisme : le jeu d'écriture utilisé pour établir un acte de naissance, un certificat ou un diplôme, les panneaux publicitaires, la maquette de présentation des journaux, les dessins sur les murs, la plaque d'information d'une église, la maquette du logo d'un lycée, les banderoles d'affichage et les exemples comme ceux-ci sont indénombrables.

Voici quelques images qu'on rencontre le plus souvent dans les rues et qui renvoient au graphisme.



Un acte de naissance



Panneau publicitaire d'un salon de coiffure



Plaque d'un établissement scolaire



Dessins artistiques sur un mur

Avec l'avancée des technologies aujourd'hui, le graphisme ne se pratique presque plus manuellement. Elle se fait avec l'appui d'outils informatiques, l'utilisation des logiciels spécialisés, facilitant le travail de l'infographe. Voici quelques produits de l'infographie qu'on rencontre couramment



Un logo d'une entreprise



Page d'accueil d'un journal



Un logo ayant une devise



Panneau publicitaire



L'infographie ne peut être dissociée du graphisme car avant toute réalisation à l'aide de la machine, l'infographe doit tout d'abord concevoir à la main sur du papier, une esquisse de sa représentation. Elle fait donc appel aux compétences d'un graphiste et à la créativité.

3. Domaines d'application de l'infographie

L'intérêt de l'infographie s'impose si on considère que l'image permet un plus grand débit d'information que le texte ; car elle est comprise globalement et non séquentiellement par l'être humain. Elle trouve ses applications dans presque tous les domaines de la création, de la conception ou de la production de l'information :

- ✚ Dans le domaine de la **publicité**, elle permet la réalisation des panneaux publicitaires (entreprises, écoles, églises, salons de coiffure, hôpitaux, pharmacies, ...), des logos des organisations.
- ✚ Dans le domaine de l'**audiovisuel**, elle permet la réalisation des bandes d'information défilant en dessous de l'écran, les séquences publicitaires télévisées ou radiodiffusées.
- ✚ Dans le domaine de la **presse écrite**, elle permet le montage des pages des journaux, des magazines, des revues, des livres, ...
- ✚ En **architecture**, on peut représenter des bâtiments avec tous leurs détails; on peut visionner ces bâtiments avec n'importe quel point de vue.
- ✚ En **cartographie**, l'ordinateur peut produire très rapidement des cartes avec différents types d'éléments géographiques.
- ✚ Dans le domaine des **transports**, permet de représenter les panneaux du code de la route.
- ✚ Dans le domaine du **cinéma**, l'infographie est à la base de la création des effets spéciaux, des bandes dessinées, des mondes futuristes et des interfaces imaginaires.
- ✚ Dans le domaine des **Visualisations scientifiques**, l'infographie permet de mieux comprendre les données ou modèles, de mieux cerner les comportements, de simuler des actions avant de les poser et même de faciliter la formation.
- ✚ En **médecine**, l'infographie médicale permet des traitements (chirurgie, radio, etc.) assistés par ordinateur et ainsi réduit les risques médicaux.

II. LES ÉLÉMENTS D'INFOGRAPHIE

I. Quelques logiciels d'infographie

Il existe de nombreux logiciels d'infographie. Voici une liste non exhaustive des produits en Français que l'on peut trouver :

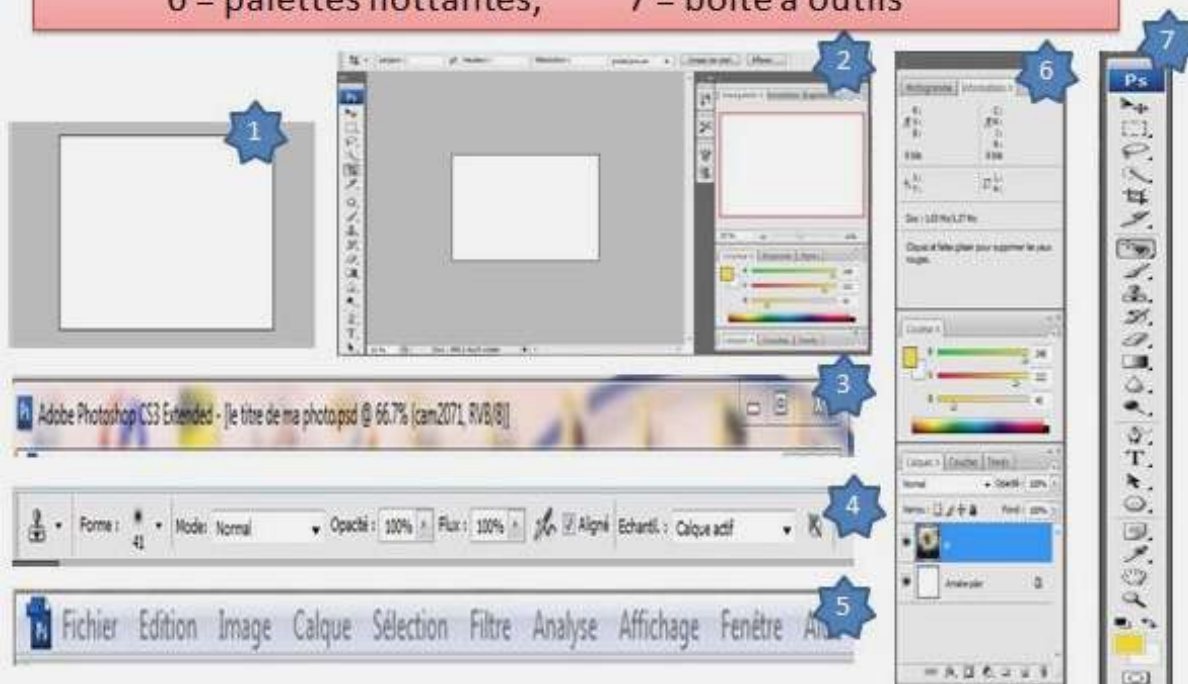
- | | |
|----------------------------------|-----------------------------|
| ✍ Adobe Photoshop | ✍ Photo DeLuxe (Adobe) |
| ✍ IPhotoExpress (U-Lead) | ✍ PhotoExpress (U-Lead) |
| ✍ Ixla Artist (ISR) | ✍ PhotoSuite (MGI Software) |
| ✍ Paint | ✍ PhotoWorks (Extrafilm) |
| ✍ Paint Shop Pro (Jasc Software) | ✍ Photo House (Corel) |
| ✍ Picturama (Hi-Image) | ✍ Photopaint |
| ✍ Picture Publisher (Micrografx) | ✍ Photo Impact (U-Lead) |
| ✍ Picture It (Microsoft) | ✍ Photo Studio (Arc Soft) |

REMARQUE

Dans un logiciel de dessin assisté par ordinateur, les icônes des outils ne sont pas toujours les mêmes. Pour avoir donc le nom ou la description d'un outil, il suffit de pointer tout simplement le curseur dessus. La plupart de nos activités s'appuiera sur le logiciel Adobe Photoshop CS3 qui nous offre des caractéristiques presque standards de tout logiciel d'infographie.

2. Interface graphique d'un logiciel d'infographie

1= zone de travail ; 2= espace de travail ; 3= barre de titre ;
4= barre d'options des outils ; 5= barre des menus ;
6= palettes flottantes ; 7= boîte à outils



L'interface d'un logiciel d'infographie dispose de différentes parties en fonction de son niveau de performance. On retrouve généralement les éléments suivants:

- ⊕ **La Barre de titre** : cette barre contient le nom du logiciel et le nom du document en cours de création (dans certains types d'affichage, ceci n'est pas toujours le cas).
- ⊕ **La Barre des menus** : cette barre présente une série de boutons qui nous permettra entre autres d'ouvrir, enregistrer, fermer, éditer ou modifier un document.
- ⊕ **La Boîte à outils** : cette fenêtre constitue notre stock d'outils, autrement dit la trousse dans laquelle on trouve les outils tels que la paire de ciseaux, les pinces, les crayons de couleur, la gomme, etc.
- ⊕ **La Barre d'options des outils** : les fonctionnalités de cette barre permettent de définir une ou plusieurs options spécifiques aux outils de notre logiciel afin de les paramétrer.
- ⊕ **Les Palettes flottantes** : ce sont des petites fenêtres que l'on dispose à l'intérieur de l'espace de travail et à partir desquelles nous aurons accès à la majorité des fonctionnalités dont nous disposons. Parmi elles, on peut citer l'historique, les calques, les couleurs, le navigateur etc.
- ⊕ **La zone de travail** : c'est la feuille sur laquelle l'infographe dessine. Il s'agit d'un élément à part entière qui compose l'espace de travail, au même titre que la boîte à outil.
- ⊕ **L'espace de travail** : il peut être comparé à un bureau de travail réel. Il s'agit de l'espace dans lequel sont disposés la boîte à outils, la zone de travail et les palettes flottantes.

3. Présentation des outils de l'infographie :

Nous présentons ici quelques outils qu'on retrouve dans un logiciel d'infographie. Cette liste n'est qu'à titre indicatif, il ne s'agit aucunement d'une liste exhaustive des outils d'infographie.

NOMS	RÔLES	OPTIONS
Outils de sélection		
Outil rectangle de sélection Outil ellipse de sélection	Permet de sélectionner une zone de forme rectangulaire ou circulaire	Style : normal, fixe Contour progressif
Outil lasso Outil lasso polygonal Outil lasso magnétique	Permet de sélectionner une zone manuellement par un cliquer-glisser	Type de sélection Contour progressif
Outil sélection rapide Outil baguette magique	Sélectionner une zone de même couleur	Forme de la sélection Type de la sélection
Outil tranche Outil sélection tranche	Séparer un objet ou une image en plusieurs parties distinctes	Style : normal, fixe
Outils de correction		
Outil correcteur de tons direct Outil correcteur Outil pièce	Permet d'apporter les corrections diverses à un objet Permet de corriger l'effet œil rouge sur une photo	Forme, Mode, Source ; Taille de la pupille, Taux d'obscurcissement

Outil œil rouge		
Outil tampon de duplication Outil tampon de motif	Permet de reproduire une zone précise d'une image ou un motif dans une autre zone de la même image	Forme, Mode, Opacité, Flux, Motif
Outil gomme Outil gomme d'arrière plan Outil gomme magique	Permet d'effacer une partie de l'image	Forme, Mode, opacité, Flux
Outil goutte d'eau Outil netteté Outil doigt	Permet de rendre une image plus ou moins nette ou floue	Forme, Mode, Intensité
Outil densité – (moins) Outil densité + (plus) Outil éponge	Permet d'améliorer la densité d'une image	Forme, Mode, Flux
Outils de forme		
Outil rectangle Outil rectangle arrondi Outil ellipse Outil polygone Outil forme personnalisée	Permet de dessiner des formes particulières dans la zone de travail	Calque de forme Tracés Pixels de remplissage Style Couleur
Outils de texte		
Outil texte	Permet de saisir un texte au clavier	Orientation, type police, taille police, alignement, la couleur, effets, style
Outil annotations	Permet de faire des annotations sur un objet	Auteur Taille, Couleur
Outils de peinture		
Outil dégradé Outil pot de peinture	Permet de faire de remplir un objet avec une couleur particulière ou un motif	Mode, Opacité, Dégradé, Tolérance, couleur
Outil pipette Outil échantillonnage de couleur	Permet de prélever une couleur particulière	Taille
Outils de dessin à main levée		
Outil pinceau Outil remplacement de couleur Outil crayon	Permet de peindre ou de dessiner avec une couleur particulière	Forme, Mode, Limites, Tolérance
Outil Zoom		
Outil zoom	Permet de réduire ou d'agrandir tout ou une partie d'un objet ou une image	
Outil de déplacement		
Outil déplacement	Permet de déplacer un objet, image ou texte sur la zone de travail	Groupe ou calque

III. UTILISATION D'UN LOGICIEL D'INFOGRAPHIE

Nous allons présenter ici des activités récurrentes effectuées sur un logiciel d'infographie. Nous allons pour cela nous appuyer sur le logiciel Adobe Photoshop CS3. C'est le logiciel professionnel le plus utilisé au monde dans le domaine de l'infographie.



ACTIVITÉ 1 : Télécharger et installer un logiciel d'infographie comme ADOBE Photoshop CS3

Le **téléchargement** et **l'installation** d'un logiciel ont déjà été vus dans les classes antérieures. Il est juste question ici de donner quelques indications supplémentaires pour facilement télécharger et installer un logiciel.

- ② Plusieurs pages web proposent l'installation d'un même logiciel. Pour les recenser, il suffit de saisir le nom du logiciel dans la zone de recherche d'un moteur de recherche. Lire les indications données par les résultats de la recherche pour chaque page proposé afin de faire le meilleur choix de la page de téléchargement.
- ② A la fin du téléchargement de votre logiciel, lancer l'installation en double-cliquant sur l'icône  apparaissant à la fin du téléchargement dans la zone où vous l'aviez enregistré. Il ne reste plus qu'à suivre les étapes classiques d'installation d'un logiciel en lisant évidemment les instructions avant de faire un quelconque choix.

Il faut noter que les logiciels sont en général proposés en version d'essai n'ayant pas besoin d'une licence mais dont l'accès est limité dans le temps et en capacité et en version complète exigeant l'achat de la licence. Faire le choix du logiciel au téléchargement ou à l'installation du logiciel selon les moyens dont vous disposez.

ACTIVITE 2 : Lancer Photoshop, créer un nouveau document et importer une image pour la retoucher

- ② Pour lancer Photoshop il suffit de double cliquer sur son icône  placée sur le bureau de l'ordinateur. On peut également cliquer sur **Démarrer**, puis sur **Tous les programmes** et enfin sur **Photoshop CS3**.
- ② La création d'un nouveau fichier se fait à partir du menu **Fichier** → **Nouveau**. A ce niveau, on donne le nom et les paramètres de notre zone de travail (largeur, hauteur, résolution, mode, contenu de l'arrière plan etc.), et valider par **OK**.
- ② Pour importer une image à partir d'un dossier, il suffit de cliquer sur **Fichier**, puis sur **Importer**. Sélectionner une image dans l'un de vos dossiers dans la fenêtre qui s'ouvre et valider en cliquant sur **Importer**. Votre image va s'afficher dans la zone de travail de votre logiciel d'infographie.

ACTIVITÉ 3 : Enregistrer un document après le travail

Pour enregistrer un document, il faut savoir sous quel format de fichier on souhaite l'enregistrer. Photoshop offre divers types de fichier image (JPG, PNG, PDF, PSD etc.). Pour enregistrer votre document il suffit de cliquer sur **Fichier** puis sur **Enregistrer / enregistrer sous**. Entrer dans la fenêtre qui s'ouvre, entrer **le nom du document**, **le format de fichier**, **le répertoire de destination** et valider avec le bouton "ENREGISTRER". L'enregistrement sous forme d'un projet qui pourra être modifié plus tard si l'on le souhaite se fait avec l'extension ".PSD*" ou "*.PDD*".

ACTIVITÉ 4 : Insérer un élément dans une page web

L'insertion d'éléments (images, boutons, graphique, texte, etc.) dans une page web à partir d'un logiciel d'infographie comme Photoshop CS3 peut se faire de plusieurs manières. Nous ne présentons dans cette partie que deux méthodes simples à utiliser.

(i) Méthode 1

Après avoir traité un élément dans la zone de travail, comme la création des boutons ou la retouche d'une image, au moment de l'enregistrement cliquer sur **Fichier**, puis sur **Enregistrer pour le web et les périphériques**. Une fenêtre s'ouvre offrant des options de présentation de votre élément dans la page web. Après avoir fait vos choix, valider en cliquant sur **Enregistrer**. Puis Enregistrer normalement votre document et donner le nom du fichier html qui contiendra vos éléments. Un dossier "images", contenant les boutons, images ou autres est créé automatiquement dans le répertoire où ils ont été enregistrés.

(ii) Méthode 2

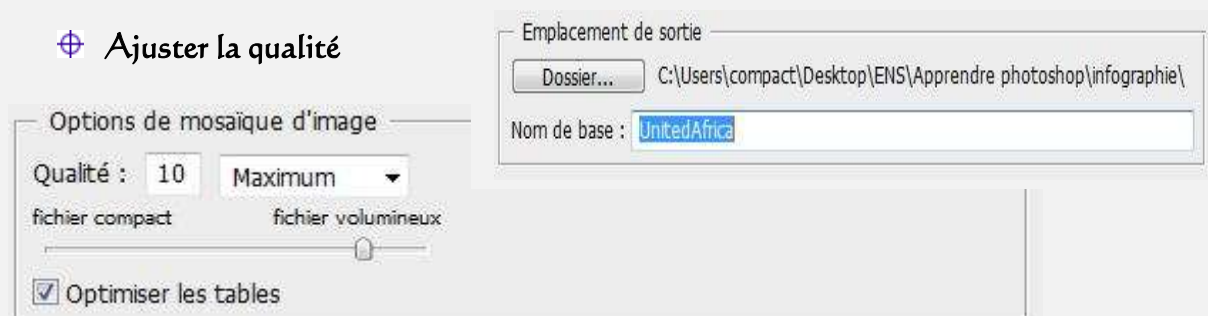
Pour insérer un élément dans une page web, on peut également procéder de la manière suivante :

Cliquer sur **Fichier**, puis sur **Exporter** et après sur **Zoomify...** Une fenêtre nommée **Exportation Zoomify** s'ouvre vous permettant de fixer les caractéristiques de votre élément et la manière dont elle sera présentée dans la page web. Faire les modifications nécessaires :

⊕ Sélectionner un dossier de sortie

⊕ Donner un nom de base

⊕ Ajuster la qualité



⊕ Ajuster l'affichage dans le navigateur web

Options du navigateur

Largeur : 600 pixels Hauteur : 600 pixels

☒ Ouvrir dans le navigateur Web

A la fin, valider en cliquant sur **O.K.**



Résumé

Infographie = Information + Informatique + Graphisme = L'Information par l'image.

L'information est une séquence d'évènements. On a plusieurs représentations de l'information :

- Le **texte** : c'est un ordre d'apparition des lettres de l'alphabet.
- Le **son** : c'est une succession d'ondes sonores.
- L'**image** : c'est un voisinage ou ensemble géométrique des pixels.
- L'**animation** : c'est un voisinage géométrique et temporel de pixels i.e. un ensemble de pixels s'étendant dans le temps.

L'**infographie** est donc une branche de l'informatique qui permet de transmettre une information par une représentation et dont l'information dépend de modifications qui sont faites sur ladite représentation à partir d'un ordinateur.

L'utilisateur des logiciels d'infographie s'appelle un **infographe** ou un **infographiste**.

L'infographie trouve ses applications dans tous les domaines de la société (architecture, cartographie, transport, astronomie, jeux vidéo, visualisations scientifiques, médecine, cinéma, publicité, etc.)

L'infographie ne peut être dissociée du graphisme car avant toute réalisation à l'aide de l'ordinateur, l'infographe doit tout d'abord concevoir à la main sur du papier, une esquisse de sa représentation.



Exercice 1 :

On désire créer une école primaire dans votre village. Le chef du village lance un concours primé pour la conception d'une plaque contenant des informations sur ladite école.

3. Quelles sont les informations qu'on peut retrouver sur une telle plaque ?
4. Faites une esquisse de cette plaque sur une feuille de papier.

Exercice 2 :

A l'occasion de la fête des mères, vous désirez concevoir à la main une carte de vœux pour votre maman.

1. Écrire de cinq (05) façons différentes l'expression suivante :
« HAPPY MUM'S DAY MUMMY »
2. Faire un dessin significatif pour votre maman et rédiger un texte qui accompagnera votre image.
3. A partir des questions précédentes, faites une maquette de cette carte de vœux.

Exercice 3 :

Concevoir la maquette de la première page du journal de votre établissement.

Exercice 4 :

1. Définir : Infographie, Graphisme, Texte, Son, Image, Animation.
2. Citer cinq logiciels d'infographie en différenciant les logiciels gratuits des logiciels propriétaires.
3. Donner les différentes représentations de l'information.
4. Citer cinq domaines d'application de l'infographie en justifiant vos réponses.
5. Citer cinq éléments de l'interface d'un logiciel d'infographie en donnant chacun sa fonction.
6. Citer dix outils d'un logiciel d'infographie en donnant chacun sa fonction.

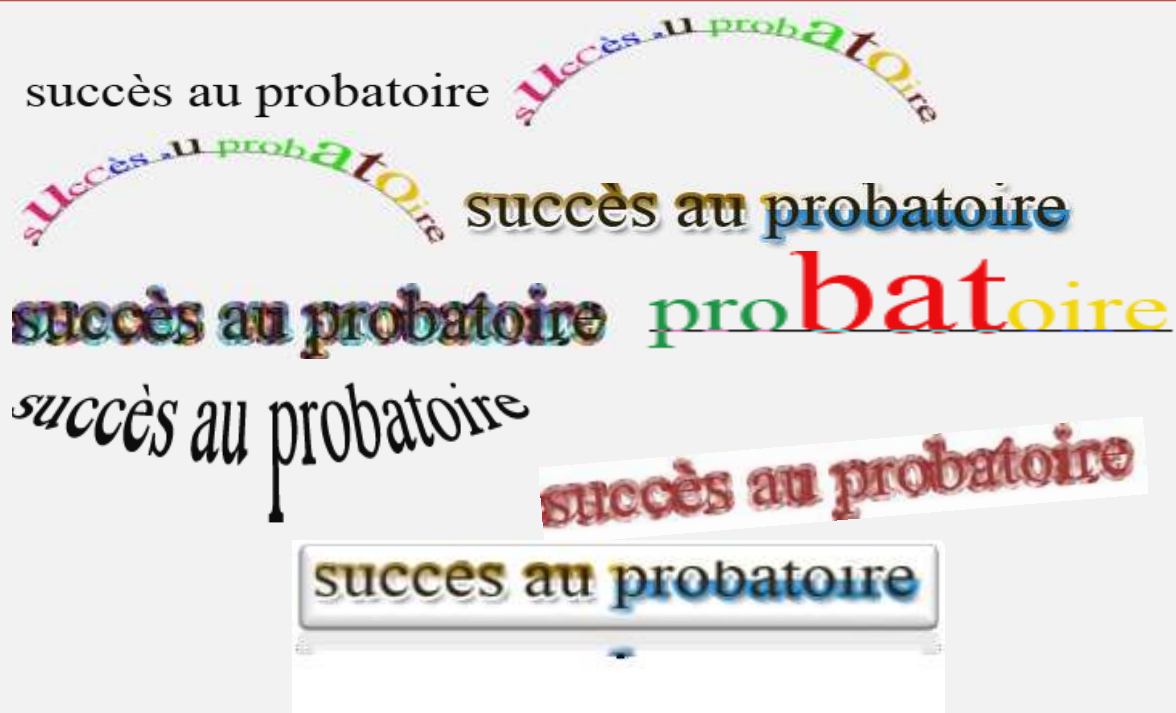
Exercice 5 :

Que représentent les éléments 1, 2, 3, 4 et 5.



Chapitre 2

CRÉATION DES TEXTES AVEC EFFET



SOMMAIRE



- I- PRÉSENTATION DES OUTILS DE TEXTE
- II- DÉFORMATION D'UN TEXTE
- III- AMÉLIORATION DE LA PRODUCTION DES DOCUMENTS
DE PAO



Problème

Le proviseur de votre lycée désire créer un logo pour valoriser et signaler le déroulement des examens du PROBATOIRE dans le lycée. Vous décidez pour le faire de former un groupe avec vos amis. Il vous est confié dans le groupe la tâche de proposer différentes formes de représentation du mot PROBATOIRE.

5. Comment procéderez-vous pour réaliser cela ?
6. Proposer différents modèles.



Pré-requis

- Savoir ce qu'est un texte ;
- Savoir écrire ou saisir un texte ;
- Savoir sélectionner un texte.



Objectifs

- Connaître les différentes modifications sur un texte ;
- Connaître les fonctionnalités de l'outil texte ;
- Déformer un texte ;
- Améliorer un document PAO.

INTRODUCTION

Dans la vie de chaque jour, on retrouve des documents (cartes de vœux, mariage, anniversaire, exposé, etc.) ayant dans leur contenu des textes avec des formes et effets divers. Ces effets ont pour but d'embellir et de mettre en exergue une information. Les logiciels d'infographie nous donnent les outils facilitant la création et la modification des effets sur un texte. De façon générale un texte a une forme, une taille, une couleur, une orientation qu'on peut modifier à son gré pour créer un effet particulier. Ces caractéristiques peuvent être combinées pour obtenir des résultats plus attractifs.

I. PRÉSENTATION DES OUTILS DE TEXTE

1. La barre d'option de l'outil de texte

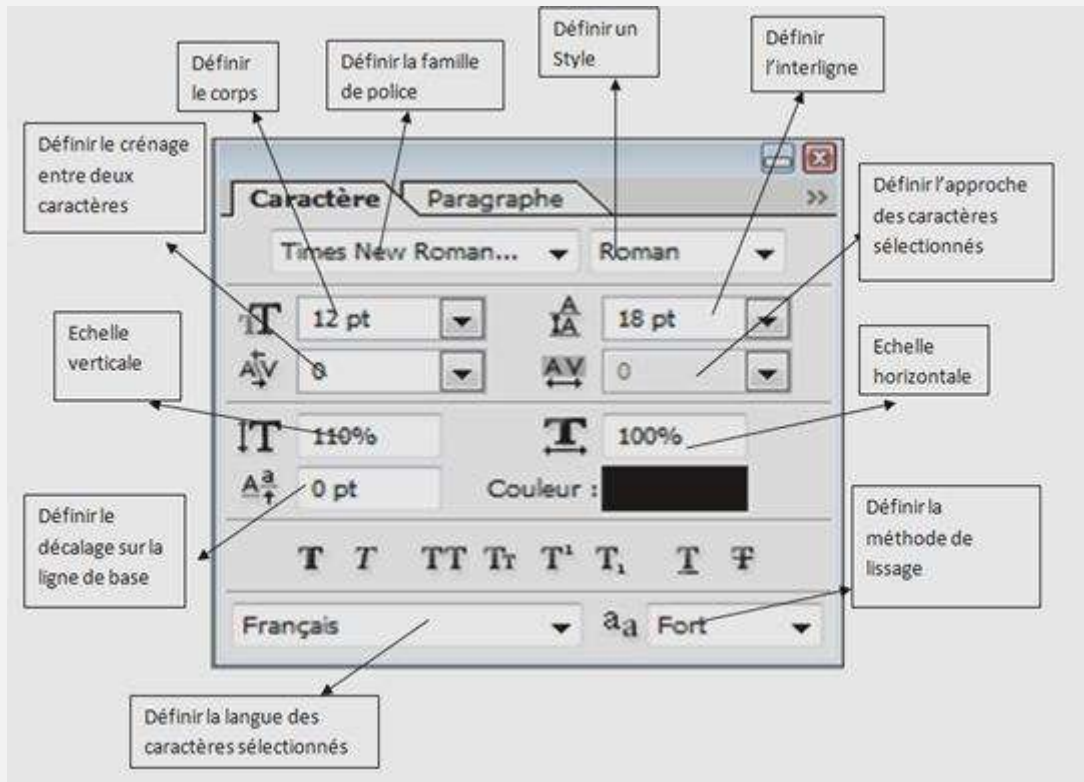
Vous pouvez créer du **texte vertical ou horizontal** à tout emplacement sur une image. En fonction de la manière dont les outils de texte sont utilisés, vous pouvez entrer un **texte de point** ou un **texte de paragraphe**. Le texte de point permet de saisir un mot ou une ligne de caractères, alors que le texte de paragraphe permet de saisir et de mettre en forme le texte avec un ou plusieurs paragraphes. L'icône de l'outil texte est la suivante **T**.



Présentation de la barre d'options de l'outil Texte

2. La palette caractère

La palette de caractère renferme des options permettant d'effectuer un ensemble d'opérations sur un texte tel que changer la couleur, la taille, l'orientation, etc. Il se présente comme suit :



Présentation de la palette caractère

3. Création du texte

Lors de la création d'un texte, il faut au préalable créer un nouveau document ou ouvrir un document existant sur lequel le texte sera saisi. Une fois un document ouvert, on procède comme suit :

- ② Sélectionnez *l'outil Texte Horizontal ou vertical* dans la palette d'outils
- ② Cliquez dans la zone de travail du document puis saisissez le texte.



On peut également insérer un texte sur une image

II. DÉFORMATION D'UN TEXTE

Déformer un texte c'est lui appliquer une forme particulière bien définie. Cette opération est très utilisée dans la vie pratique : dans la réalisation des cartes de vœux, des publicités, pour la création des logos, des couvertures des livres, etc. A cet effet, on peut utiliser aussi bien les logiciels de traitement d'image (Photoshop, Paint ...) que les logiciels de traitement de texte (Word, OpenOffice ...).

I. Quelques styles de déformation

Il existe plusieurs modèles standards de déformation. Comme exemple de style de déformation nous avons :

@ Le style Arc

Comme son nom l'indique il déforme le texte sous forme d'arc. C'est un style qu'on retrouve le plus souvent sur les logos ayant une forme cyclique, sur les cartes de vœux ayant des images sous forme de cercle.

Exemple :



@ Le style Renflement

Ce style de déformation renfle le texte. On le retrouve le plus souvent sur les plaques de publicité.

Exemple :



2. Procédure de déformation d'un texte

Après avoir créé le texte, on peut le déformer. Pour le faire on peut procéder de trois façons :

1^{ère} façon de faire :

- @ Faire un **clic droit** sur le texte saisi.
- @ Choisir **déformer le texte...** dans le menu contextuel. Une fenêtre s'ouvre ayant pour nom **Déformer le texte**
- @ Choisissez un **style de déformation** dans le menu déroulant **Style** et cliquez sur **OK**.

2^{ème} façon de faire :

- @ Cliquez d'abord sur le menu **Calque** de la barre de menu ensuite sur **Texte** et enfin sur **Déformer le texte**. On obtient la même fenêtre que plus haut.
- @ Choisissez un **style de déformation** dans le menu déroulant **Style** et validez sur **OK**.

3^{ème} façon de faire :

- @ Cliquez sur l'outil **créer un texte déformé** sur la barre d'option de l'outil Texte
- @ Choisissez un **style de déformation** dans le menu déroulant **Style** et cliquez sur **OK**.

3. Modifier la forme, la couleur, la taille, l'orientation

Après avoir saisi un texte, on peut vouloir changer son apparence, soit en modifiant la couleur, la taille... Pour s'y prendre, on peut procéder comme suit :

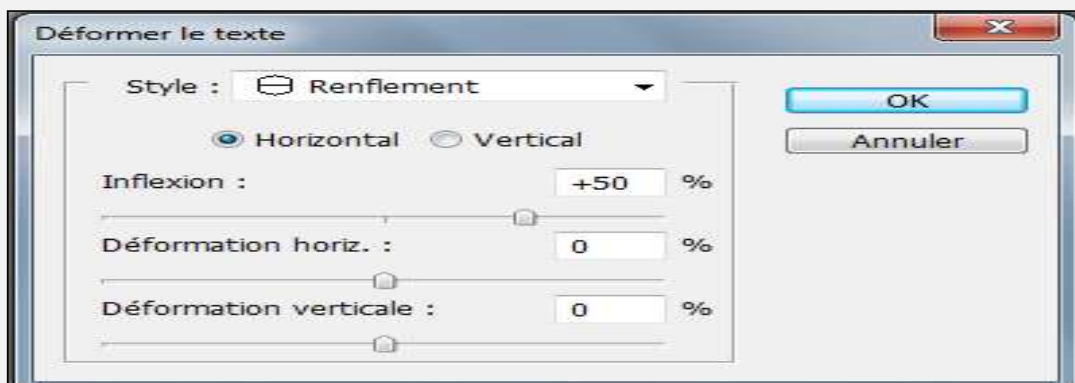
- ⓐ Sélectionnez le texte sur lequel on veut appliquer des modifications
- ⓐ Cliquez sur l'onglet **Caractère** pour amener la **palette Caractère** au premier plan ; Si cette palette n'est pas ouverte choisissez **le menu Fenêtre** puis **Caractère**
- ⓐ choisir **la taille**, la **famille de police** suivit de son **style**, la **couleur** et bien d'autres paramètres selon ce qu'on veut faire.



PROTOCOLE DE DEFORMATION DU TEXTE ET CHANGEMENT DE POLICE, COULEUR, TAILLE

Étape 1 : Écrire le mot **PROBATOIRE** dans la zone de travail d'un nouveau document.

Étape 2 : Faire un clic droit sur le texte, choisir **déformer le texte...** dans le menu contextuel. On obtient la fenêtre ci-dessous



Étape 3 : Choisir le style de déformation **ARC** dans le **menu déroulant Style** et valider.

Étape 4 : Ouvrir la palette de caractère

Étape 5 : Sélectionner les trois premières lettres du mot, cliquer sur l'option **couleur** et choisir la **couleur verte**, puis dérouler l'option **taille** et choisir **30pt**. Sélectionner les trois lettres suivantes leur appliquer **couleur rouge** et **taille 60pt** et le reste des lettres en **couleur jaune** avec une **taille 36pt**

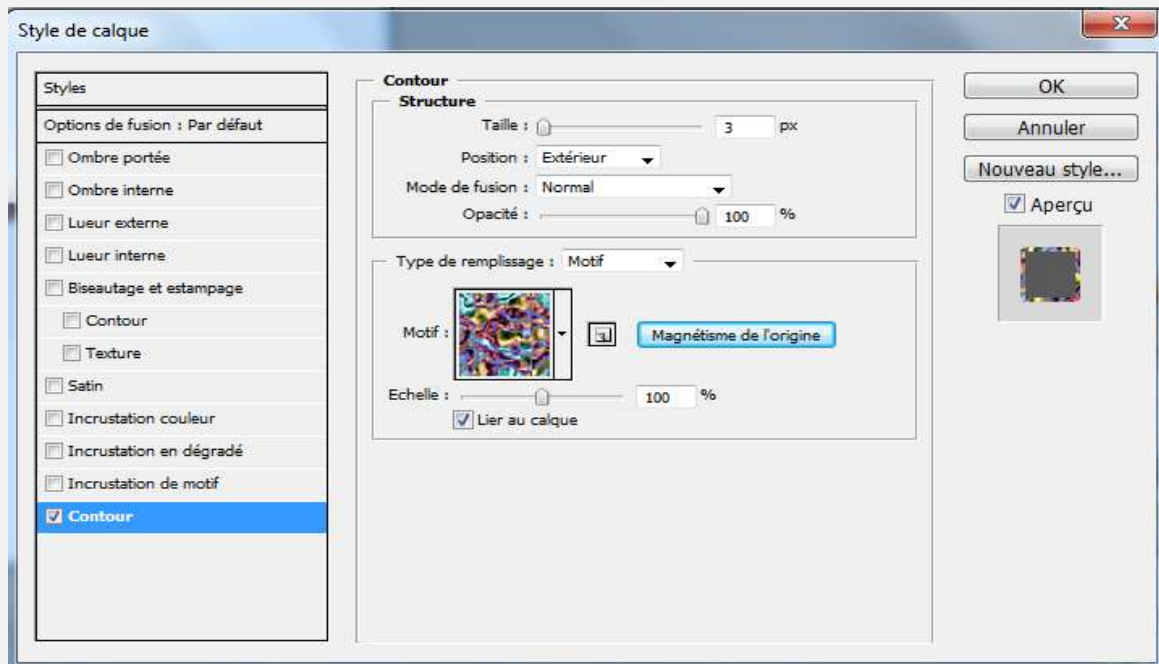
Le résultat obtenu en fin de compte est :



4. Ajuster les angles et les contours et ajouter des effets au texte et les affiner

Après avoir créé un texte on peut vouloir y ajuster les angles et les contours, on peut également vouloir lui ajouter des effets. Ceci est dans le but de rendre ce texte plus beau ou beaucoup plus présentable. Pour ce fait nous allons procéder de la manière suivante, après avoir créé le texte:

- ② Sélectionnez le **menu Calque > Style de calque > Biseautage et estampage ou contour**. La fenêtre obtenue est la suivante :



- ② A ce niveau on peut choisir un **style** (ombre, contour, texture, etc. ...) et le configurer à notre convenance.



PROTOCOLE DE CREATION DU TEXTE EN AJUSTANT LES CONTOURS

Étape 1 : Ecrire le texte **SUCCES AU PROBATOIRE**

Étape 2 : Sélectionnez le menu **Calque → Style de calque → contour...** Vous allez obtenir une fenêtre semblable à celle-ci-dessus.

Étape 3 : Au centre dans le cadre intitulé structure, mettre **la taille à 5, position extérieur, mode de fusion normal, opacité 100%, type de remplissage motif** (dérouler la liste déroulante qui s'y trouve et choisir un motif qui vous convient) **et échelle 100%** et valider sur **OK**. Le résultat sera le suivant :

succès au probatoire



Toutes ces modifications peuvent être apportées à partir de la palette d'outil.

III. AMÉLIORATION DE LA PRODUCTION DES DOCUMENTS DE PAO (BILLET D'INVITATION)

Dans cette partie nous allons monter un billet d'invitation. Cette façon de faire n'est pas standard. Nous allons produire un billet d'invitation d'anniversaire



PROTOCOLE DE CREATION D'UN BILLET D'INVITATION

Étape 1 : Lancer l'application Photoshop et ouvrir un nouveau dossier de 600 pixels de hauteur et de 300 pixels de largeur.

Étape 2 : Importer une image pour le fond du papier. Pour cela on procède comme suit : fichier/ importer/ choisir le dossier dans lequel se trouve votre image et choisir l'image qui vous convient et valider sur OK. Une fois l'image importé, l'étendre sur toute la zone afin de tout couvrir le fond puis double cliquer dessus pour valider. Après avoir importé une image un nouveau calque est automatiquement créer. Voir l'image ci-contre



Étape 3 : Importer une autre image significative par rapport à l'évènement. Procéder de la même manière qu'à l'étape 2 pour importer l'image. Une fois l'image importer le redimensionner afin qu'il n'occupe qu'une petite zone à gauche de la carte puis double cliquer. Un nouveau calque est créé. Voir l'image ci-contre.



Étape 4 Importer une autre image significative par rapport à l'évènement. Procéder de la même manière qu'à l'étape 2 pour importer l'image. Une fois l'image importer le redimensionner afin qu'il n'occupe qu'une petite zone à droite et en bas de la carte puis double cliquer. Un nouveau calque est créé. Voir l'image ci-contre



Étape 5 saisissez l'invitation proprement dit. Vous pouvez saisir le mot suivant « Invitation » dans son calque. Puis le texte suivant : « Avoir un an de plus c'est... » dans un autre calque avec une taille de police plus petite que celle du mot « Invitation ».



Résumé

Plusieurs opérations peuvent être appliquées sur un texte après sa **création**. On peut soit le **déformer**, cela revient à lui appliquer une forme particulière bien définie. Comme forme nous avons **le Style Arc** qui se présente sous forme d'arc, le **Style Poisson** qui a la forme d'un poisson, **le style Drapeau** qui se présente sous forme de drapeau flottant dont on a pris une photo, etc. Trois procédures permettant de déformer un texte ont été énumérées parmi lesquelles la suivante :

- ④ *Faire un clic droit sur la zone de texte*
- ④ *Choisir « déformé le texte... » sur le menu contextuel. Une fenêtre s'ouvre ayant pour nom « Déformer le texte »*
- ④ *Choisissez un style de déformation dans le menu déroulant Style et cliquez sur OK.*

Comme opérations appliquées à un texte, on peut changer sa **couleur**, de même que sa **police**, **la taille de la police**, changer **son orientation**, etc. Pour effectuer ces différentes tâches on peut soit utiliser *la palette de caractère* soit *la barre d'outils texte horizontale*. On peut aussi ajuster les **angles et contours** d'un texte et de même que lui **ajouter des effets**.

Ces différentes opérations sont très utilisées dans la vie pratique : dans la réalisation des cartes de vœux, des publicités, pour la création des logos, des couvertures des livres, etc.



NB : Réaliser chacune de ces tâches dans un premier temps à la main, et ensuite à l'aide d'un ordinateur.

EXERCICE 1:

Écrire un mot et appliquer à chaque lettre de ce mot une couleur qui lui est propre, et changer la taille.

EXERCICE 2:

Créer un texte de trois mots minimum et cinq au maximum et appliquer à chacun de ces mots un contour précis.

EXERCICE 3:

Créer un texte de trente mots minimum, applique à chaque caractère une couleur proche de celle du caractère voisin (mais différent) de tel sorte qu'en lisant le texte on n'ait pas la sensation de changer de couleur.

EXERCICE 4:

Créer un texte d'une ligne au maximum et le rendre vertical.

EXERCICE 5:

Créer une carte de vœux en combinant toutes les notions vue dans ce chapitre.

Chapitre 3

RETOUCHE D'UNE PHOTO ET CRÉATION DES BOUTONS



SOMMAIRE



I- RETOUCHE D'UNE IMAGE

II- CRÉATION DES BOUTONS ET LOGOS



Problème

Les images sont dans notre contexte une façon de ramener dans le présent le passé, une façon de se remémorer les moments forts. Il arrive parfois que ces images contiennent des imperfections (coloration, décor, yeux rouges,...) ou que l'on veuille y ajouter une touche particulière et personnelle. Instinctivement, on se demande comment on va y arriver ? Comment va-t-on modifier nos images pour obtenir des images dénuées d'imperfections ? Comment va-t-on truquer une photo, ajouter des éléments ? Et même, comment réaliser nos propres logos et boutons ?



Pré-requis

- Importer un fichier image ;
- Savoir utiliser un logiciel d'infographie
- Connaître quelques formats des fichiers image



Objectifs

- Connaître les modifications applicables sur une image
- Modifier ou truquer une photo (ajouter du texte, changer l'arrière plan)
- Extraire une partie d'une photo
- Fusionner plusieurs photos
- Créer des boutons
- Créer des objets complexes (logos, décors)

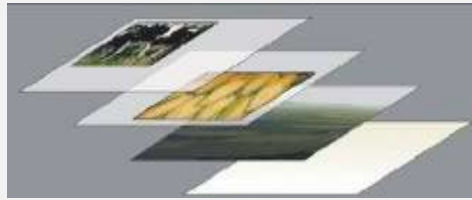
INTRODUCTION

Retoucher une photo ou une image consiste à lui apporter des modifications soit en ajoutant des éléments, soit en les retirant. Dans chacun des cas on en ressort avec une image différente de l'image initiale. Dans cette partie qui concerne la retouche d'une image, nous allons voir comment le logiciel Photoshop nous permet d'ajouter du texte à une image, de changer son arrière plan, d'extraire une partie d'une image, d'ajouter un objet dans une image et enfin de fusionner plusieurs images pour n'en faire qu'une.

I. RETOUCHE D'UNE IMAGE

1. Les calques

Ce sont des objets transparents semblables à des feuilles de dessin pouvant être empilés les uns sur les autres. Les zones transparentes permettent de visualiser les calques en dessous. Un calque peut être configuré pour laisser passer une partie des calques inférieurs (opacité, fusion...). L'image finale est formée par l'empilement des calques. La figure ci-dessous donne un aperçu de quatre calques.



On peut effectuer de multiples opérations sur les calques. On a par exemple la création, la suppression, le déplacement, la définition comme arrière plan. On peut également verrouiller un calque, le rendre visible ou invisible. On distingue plusieurs types de calque : calque image, calque d'arrière plan, calque texte, calque forme, calque réglage, calque remplissage.

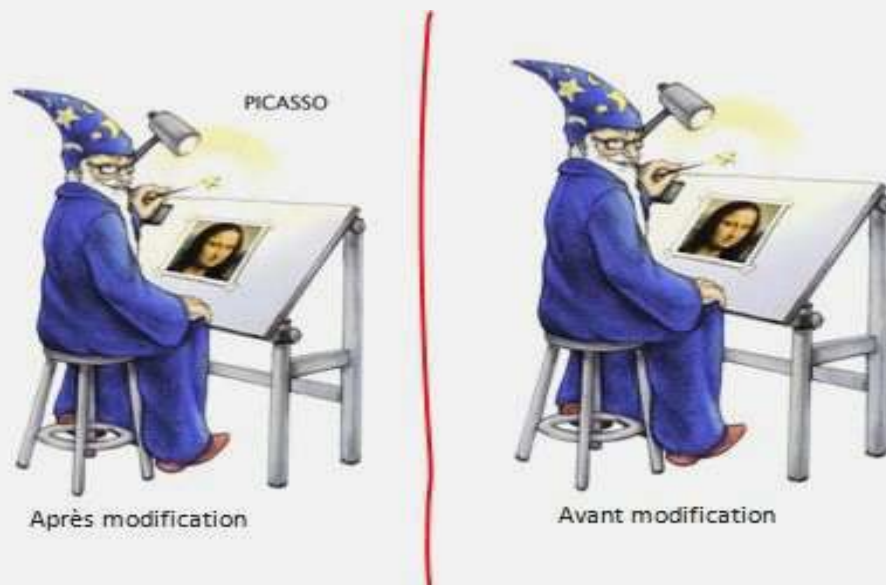
2. Ajouter du texte à une image

Pour réaliser ce travail, appelons notre peintre PICASSO. Tout d'abord lancer l'application et suivre le protocole ci-après.

④ Faire un clic droit sur l'outil texte matérialisé par le symbole suivant . Choisir une option de saisie de texte (horizontale ou verticale). Automatiquement un calque se crée et vous pouvez saisir votre texte.

④ Saisir le texte

On obtient le résultat suivant :



APPLICATION : refaire cette activité en écrivant **PICASSO** verticalement et à gauche.

3. Retoucher une image

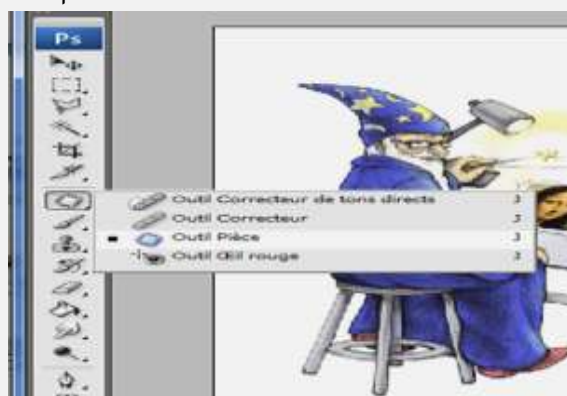
Le procédé ou protocole de retouche d'une image à l'aide de l'outil Correcteur est le suivant :

④ Lancer l'application d'infographie.

④ Importer l'image que vous voulez retoucher. Nous obtenons notre image comme suit :



② Cliquer sur l'outil **Correcteur**



- ② Définir les options. Tout d'abord, ouvrir la barre d'option en faisant **fenêtre/option**. Faire les modifications suivantes :
 - **Mode**: choisir **Remplacer**
 - **Source** : choisir **Echantillon**
 - **Aligné** : ne pas cocher **aligné**
- ② Définir le point d'échantillon : cette étape correspond en quelque sorte au choix de la couleur qui sera appliquée. Conserver la touche **Alt** enfoncée et cliquer sur une zone au choix. Nous choisissons ici la couleur **Bleu** du peintre
- ② Faire glisser l'outil dans l'image (à l'endroit voulu). Nous allons choisir de peindre en bleu deux pied de la table du peintre.

On obtient le résultat ci-après :



APPLICATION : Retoucher une image, mais cette fois utiliser l'outil pièce.

4. Déformer une image

Avec cette activité, c'est facile de faire carrière dans la caricature et devenir un bon

caricaturiste. On va se charger de déformer l'image suivante



. Le protocole est le suivant :

- ④ Lancer l'application d'infographie.
- ④ Ouvrir l'image du bonhomme dans la zone de travail.
- ④ Déverrouiller le calque s'il est verrouiller en faisant : **clik droit sur le calque/convertir en objet dynamique.**
- ④ Faire **Edition/transformation/déformation.** On peut également faire des transformations en utilisant la transformation personnalisée. Vous observez que sur l'image il ya **une grille de transformation.**
- ④ Dans la barre d'options, choisissez une **déformation** (poisson dans notre cas). Vous pouvez vous amuser à faire différentes modifications.
- ④ Enregistrer le résultat comme une image en faisant **Fichier/enregistrer sous.** Dans la boîte de dialogue qui s'ouvre spécifier le nom et le format de l'image.
On obtient l'image ci-contre :



APPLICATION : Ouvrir une image et la déformer en à votre guise

5. Ajouter un objet dans une image

Cette activité est très intéressante dans la mesure où elle vous permet de truquer des images. Pour le faire il faut utiliser l'objet calque. Pour une meilleure organisation, chaque calque doit contenir un seul objet. Puis il faudra superposer les calques dans le bon ordre pour obtenir l'image voulue. Elle peut également vous servir pour modifier l'arrière plan ou le décor d'une image

Donnons nous comme travail d'ajouter au pied de notre peintre Picasso le bonhomme suivant qui a été préalablement déformé. On suppose au départ que vous avez l'image du bonhomme déformé dans un calque que vous nommez **bonhomme**. La procédure est la suivante :



- ④ **Déverrouiller** les calques. On le fait dans le but de pouvoir déplacer les calques
- ④ Créer un nouveau calque en faisant : **Calque/nouveau/calque**
- ④ Dans la fenêtre qui s'ouvre, inscrire juste le nom du calque : **Picasso**

- Ⓢ Dans la palette calque cliquer sur le calque nommé **Picasso**. Ceci vous permet de vous situer dans ce calque que vous pouvez considérer comme une feuille de dessin pour faire des modifications
- Ⓢ Il faut maintenant inscrire l'image de Picasso en faisant simplement : **fichier/importer**. La boîte de dialogue qui s'affiche vous permet de rechercher l'image de Picasso sur la base que vous connaissez le chemin d'accès absolu.
- Ⓢ Dans la palette calque, glisser le calque Picasso en dessous de celui de bonhomme à tel enseigne qu'en parcourant les calques de bas en haut, on ait Picasso, bonhomme
- Ⓢ **Ajuster** enfin les images sur les calques pour obtenir le résultat ci-dessous
- Ⓢ Enregistrer le résultat.



APPLICATION : Refaire cette activité en ajoutant deux objets de votre choix dans l'image.

6. Extraire un objet d'une image

Cette activité montre comment on peut retirer d'une image un objet particulier qui nous intéresse dans plusieurs buts : l'insérer dans d'autres images, le conserver, le modifier. L'extraction d'un objet de son arrière plan peut se faire à l'aide de l'outil **baguette magique** ou **sélection rapide** ou à partir du **filtre Extraire**. On peut également extraire un élément dans une image en utilisant l'outil **gomme d'arrière plan**, l'outil **gomme**, l'outil **gomme magique** pour effacer de l'image tout élément différent de celui que l'on recherche.

Protocole d'extraction d'un objet dans une image en utilisant l'outil baguette magique.

Soit à extraire la pomme du milieu (en omettant son ombre) dans l'image suivante :



- Ⓢ Lancer l'application Photoshop et importer l'image. Dans votre cas, vous choisirez une image qui vous convient.
- Ⓢ Dans la palette d'outil, choisir **l'outil sélection rapide** matérialisé par 
- Ⓢ à l'aide de cliquer-glisser, sélectionner avec délicatesse la pomme. On peut constater que la sélection n'est pas exactement celle désirée ; on peut améliorer la sélection en utilisant les options suivantes de l'outil sélection rapide 
- Ⓢ copier la sélection et la coller dans un nouveau calque.
- Ⓢ Dans notre cas on obtient l'image ci-contre :



APPLICATION : Refaire cette activité en utilisant l'outil baguette magique.

II. CRÉATION DES BOUTONS ET LOGOS

Les **boutons** peuvent être définis comme des images significatives dans ce sens qu'elles portent des textes ou des icônes ayant un sens précis. Les boutons sont conçus dans le but de faciliter la navigation dans les sites web.

Les **logos** sont des images conçues dans le but de représenter une organisation, une association, une marque, ...

Il est à noter que la créativité n'a pas une formule, c'est à force de concevoir des boutons, des logos, des images que l'on finit par développer cet aspect la créativité.

Il existe plusieurs outils qui permettent la création de ces objets à l'instar de l'outil **crayon**, l'outil de **Recadrage**, l'outil **gomme**, l'outil **forme**,...

Un protocole de création d'un bouton est le suivant :

- Ⓢ Lancer l'application Photoshop et créer un nouveau fichier en faisant : **fichier/nouveau**
- Ⓢ Dans la boîte de dialogue qui s'ouvre, spécifier le nom et les dimensions du bouton
- Ⓢ Cliquer sur l'outil forme **outil Rectangle arrondi** dans la palette d'outil matérialisé par 
- Ⓢ Dessiner dans la zone de travail le bouton
- Ⓢ Choisir dans la palette style un style approprié. Dans cet exemple nous choisissons le **style coucher du soleil**
- Ⓢ On peut y ajouter du texte. Il suffit simplement de cliquer sur l'outil texte spécifié par  , puis cliquer sur le bouton et saisir. Les

modifications de style et forme peuvent être apportées au texte saisi.
Pour le faire, voir le chapitre [créer les textes avec effet](#)

APPLICATION : Dessiner un logo dans lequel sera écrit « Paix-Travail-Patrie », on y trouvera la carte du Cameroun et le drapeau du Cameroun.

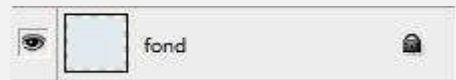


ACTIVITE : Concevoir le plan d'un site web

L'idée est de mettre sur pied un plan de site web avec les boutons de votre choix. Nous allons vous faciliter la tâche en vous donnant une maquette et les étapes à suivre pour y arriver. La maquette est la suivante :

- ⊗ Tout d'abord, fabriquer l'arrière plan en utilisant [l'outil rectangle](#).
- ⊗ Donner la couleur de fond suivante  grâce à [l'outil Pot de peinture](#)
- ⊗ Verrouiller le calque qui contient le fond grâce à l'onglet  de la [palette de calque](#)
- ⊗ Créer un nouveau calque que vous nommez [titre](#).
- ⊗ Rendre le calque de fond invisible pour pouvoir travailler sur le calque titre. Pour cela,

aller sur le calque [fond](#) et décocher l'œil



- ⊗ Utiliser les boutons préconçus ou alors fabriquer d'autres boutons qui porteront les noms des menus de votre site
- ⊗ Créer un autre calque nommé [logo](#)
- ⊗ Fabriquer un logo que vous mettrez au coin supérieur gauche du calque



Résumé

Les [calques](#) sont des objets transparents qui jouent le rôle de feuilles de dessin pouvant être empilés les uns sur les autres.

[Retoucher](#) une image ou une photo consiste à lui apporter une modification particulière grâce aux outils du logiciel utilisé.

On peut [modifier](#) une image en lui ajoutant des éléments (du texte, une image, une déformation,...) ou en retirant.

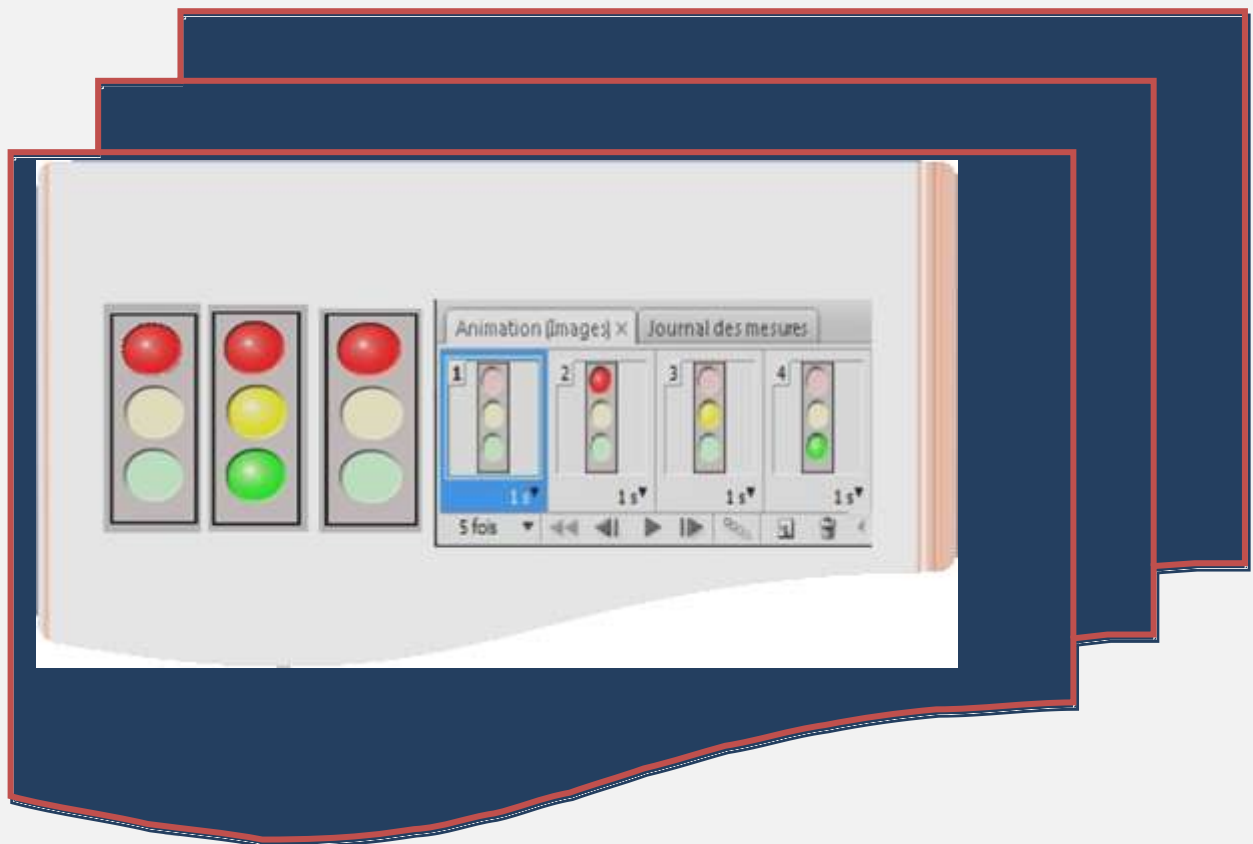
La construction des [boutons](#) ou des [logos](#) est une activité qui nécessite votre sens de créativité et un logiciel d'infographie



1. Définir : Calque, pixel, logo
2. A quoi sert un calque ?
3. Quelle est la procédure à adopter lorsqu'on doit manipuler plusieurs images pour en créer une nouvelle ?
4. Citer cinq outils qui permettent de retoucher une image et donner leur fonctionnement.
5. D'après vous, est ce qu'il faut croire aux images que l'on trouve dans les magazines.
6. Quelles sont les étapes incontournables dans la réalisation d'un bouton.
7. Concevoir le logo de votre établissement. Le réaliser à la main, puis à l'aide d'un logiciel d'infographie
8. Prendre une photo d'une personne: modifier manuellement les paramètres suivants :
 - La couleur de la chaussure
 - Ajouter une moustache
 - Ajouter les boucles d'oreilles
 - Etc.

Chapitre 4

CRÉATION DES ANIMATIONS



SOMMAIRE



- I. PRÉSENTATION DE LA PALETTE D'ANIMATION
- II. OPERATIONS FAITES DANS LA PALETTE D'ANIMATION



Problème

Votre PAPA est propriétaire d'une compagnie d'auto-école. Pour rendre sa structure plus compétitive et rechercher des partenaires internationaux, il décide de créer un site web pour sa compagnie. Sachant que votre programme d'informatique en Première comporte un module INFOGRAPHIE, il vous demande de mettre en valeur toutes vos connaissances dans ce domaine en lui fournissant une animation qu'il pourra insérer dans son site web.

- Quel type de logiciel allez vous utiliser ?
- Concevez une animation adaptée à ce site.
- Réalisez-la.



Pré-requis

- Connaître les fonctions de base d'un logiciel d'infographie
- Savoir manipuler les outils de la boîte à outil
- Savoir manipuler les calques



Objectifs

- Comprendre la notion d'animation
- Connaître et manipuler la palette d'animation
- Monter une animation simple

INTRODUCTION

Une **animation** est une séquence d'images affichées dans le temps. Chaque image diffère très légèrement de la précédente, créant une illusion de mouvement lorsque les images sont visualisées en succession rapide. Les animations permettent d'agrémenter les **pages web**.

La création d'une animation nécessite certains paramètres tels que : un ensemble d'images, un temps d'affichage défini pour chaque image. Pour réaliser une animation dans un logiciel d'infographie, il faut utiliser principalement la **palette Animation** conjointement à la **palette Calque**.

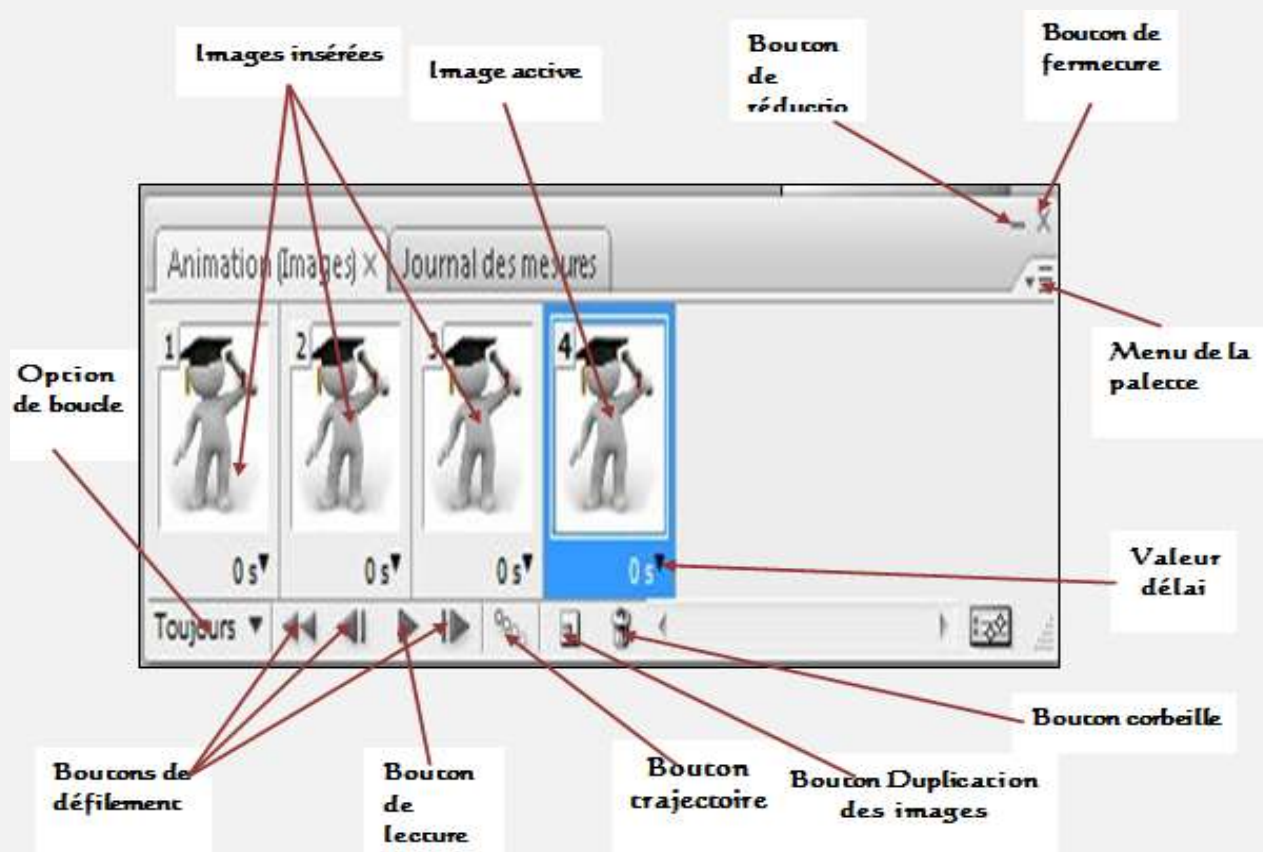
I. PRÉSENTATION DE LA PALETTE ANIMATION :

La palette Animation permet de créer, d'afficher et de définir des options pour les images d'une animation.

Affichage de la palette Animation : Choisissez **Fenêtre > Animation**. Une coche indique que la palette est activée.

Utilisation du menu de la palette Animation : Cliquez sur le bouton  dans le coin supérieur droit de la palette pour accéder aux commandes de manipulation des animations.

Modification de la taille des vignettes d'animation : Choisissez **Options de palette** dans le menu de la palette Animation, sélectionnez **une taille de vignette**, puis cliquez sur **OK**.



Présentation de la palette d'animation

II. LES OPERATIONS OFFERTES PAR LA PALETTE

1. Définition de la boucle

La **boucle** est une option qui permet de spécifier combien de fois une séquence d'animation sera répétée. Pour spécifier une boucle :

- Ⓢ Cliquer sur la case de sélection de **l'option de boucle** Toujours ▼ dans le coin inférieur gauche de la palette Animation.
- Ⓢ Sélectionner une option de boucle : **Une fois, Toujours ou Autre.**

2. Définition d'un délai pour les images

On peut définir **un délai** (temps pendant lequel est affichée une image) pour une ou plusieurs images dans une animation. Le délai est affiché en secondes. Pour définir un délai il faut :

- Ⓢ Sélectionnez une ou plusieurs images.
- Ⓢ Dans la palette Animation, cliquer sur la **valeur Délai**, 0s sous l'image sélectionnée, pour afficher **le menu contextuel**.
- Ⓢ Spécifier le délai en Choissant une valeur dans le menu contextuel, puis cliquer sur **OK**.

3. Sélection d'images

Avant de travailler sur une image, il faut la sélectionner en tant qu'**image active**. Le contenu de l'image activée apparaît dans la fenêtre du document. On peut sélectionner plusieurs images, contiguës ou non, afin de les modifier ou de leur appliquer des commandes en tant que groupe.

Pour sélectionner une image, il suffit de cliquer dessus dans la palette Animation ou dans la palette Calque.

Pour sélectionner plusieurs images contiguës, cliquez sur une première image ; tout en maintenant la touche **SHIFT** enfoncée, cliquez sur la dernière image. La dernière image et toutes les images entre la première et la dernière sont ajoutées à la sélection.

Pour sélectionner plusieurs images non contiguës, cliquez sur les images supplémentaires à ajouter à la sélection tout en appuyant sur la touche **CTRL**.

4. Réorganisation et suppression des images

On peut changer la position des images dans une animation et inverser l'ordre des images contiguës sélectionnées, supprimer des images sélectionnées ou l'animation entière.

Pour changer la position d'une image :

- ② Sélectionner l'image à déplacer dans la palette Animation
- ② Faire glisser la sélection vers la nouvelle position.

Pour inverser l'ordre des images contiguës :

- ② Sélectionner les images contiguës à inverser.
- ② Choisir **Inverser les images** dans le **menu de la palette Animation**.

Pour supprimer des images sélectionnées :

Dans la palette Animation, utiliser l'une des méthodes suivantes :

- ② Sélectionner **Supprimer** une image dans le **menu de la palette**.
- ② Cliquer sur le **bouton Corbeille**,  puis sur **Oui** pour confirmer la suppression.
- ② Faire glisser l'image sélectionnée vers le bouton Corbeille.

Pour supprimer une animation entière :

Sélectionner **Supprimer l'animation** dans le **menu de la palette Animation**

5. Ajout d'images

La première étape de la création d'une animation consiste à ajouter des images. Si une image est ouverte, la palette Animation l'affiche comme première image d'une nouvelle animation. Chaque image qu'on ajoute débute comme duplication de l'image précédente. Modifier ensuite l'image en utilisant la palette Calque. Pour ajouter une image à une animation, on dispose de plusieurs options :

- ② Dans la palette Animation, cliquer sur le bouton **Duplication des images sélectionnées** 
- ② Dans le **menu de la palette**, choisir **Nouvelle image**.

REMARQUE

- ✈ On peut utiliser la commande Trajectoire pour ajouter ou modifier automatiquement une série d'images entre deux images existantes.
- ✈ Les fonctionnalités des différentes palettes d'un logiciel d'infographie dépendent de la performance de celui-ci. On ne retrouve pas forcément tous ces boutons dans tous les logiciels d'infographie.

ACTIVITÉS

PROTOCOLE DE CRÉATION D'UNE ANIMATION : LES FEUX DE SIGNALISATION

Étape 1 : créer une nouvelle zone de travail avec les dimensions suivantes

Largeur = 110 pixels ; hauteur = 300 pixels ;

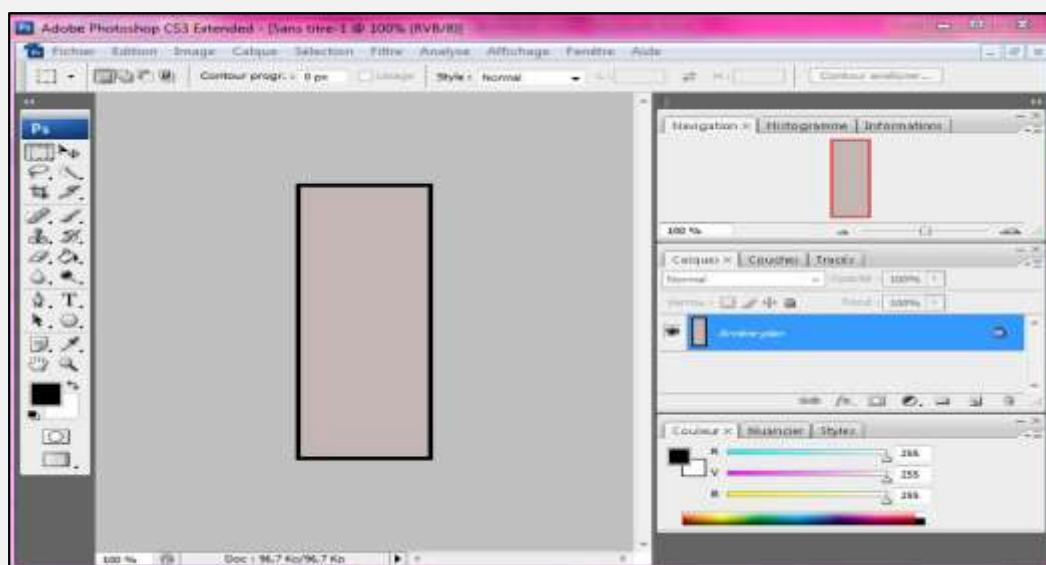
Étape 2 : Choisir une couleur de premier plan grisée. Pour le faire cliquer sur l'icône de **définition de la couleur de premier plan** et choisir la couleur dans la fenêtre de sélection la couleur et valider à **OK**.



Étape 3 : sélectionner toute la zone de travail en faisant **CTRL+A** ou menu **Sélection > Tout Sélectionner**. Pour appliquer la couleur faire la Combinaison de touche **ALT+Backspace**.



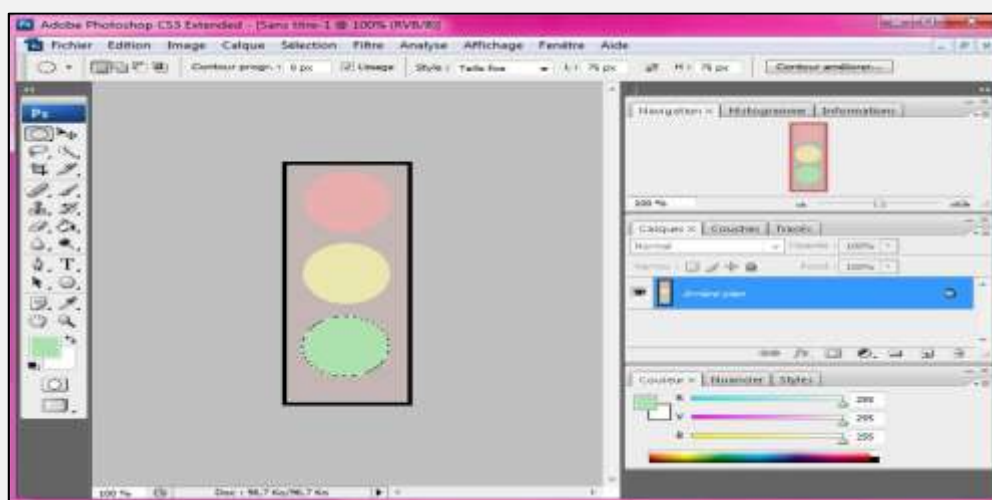
Étape 4 : rétablir la couleur de premier plan par défaut (Noire). Ensuite sélectionner à l'aide de l'outil **Rectangle de sélection**, de petites zones tout au tour de la zone de travail et appliquer la couleur par défaut en procédant de la même façon qu'à l'étape précédente. (**ALT+Backspace**)



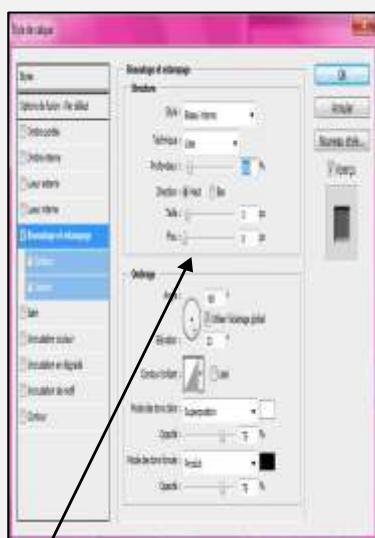
Étape 5 : Choisir l'outil **Ellipse de sélection** avec les options suivantes : **style = taille fixe ; largeur=75 px ; hauteur=75px**. Dessiner dans la zone de travail un premier rond. Modifier la couleur de premier plan en **rouge pâle** et appliquer cette couleur au cercle.

Étape 6 : Répéter la manœuvre précédente en dessinant un deuxième cercle en dessous du précédent mais en appliquant cette fois ci une couleur **jaune pâle**.

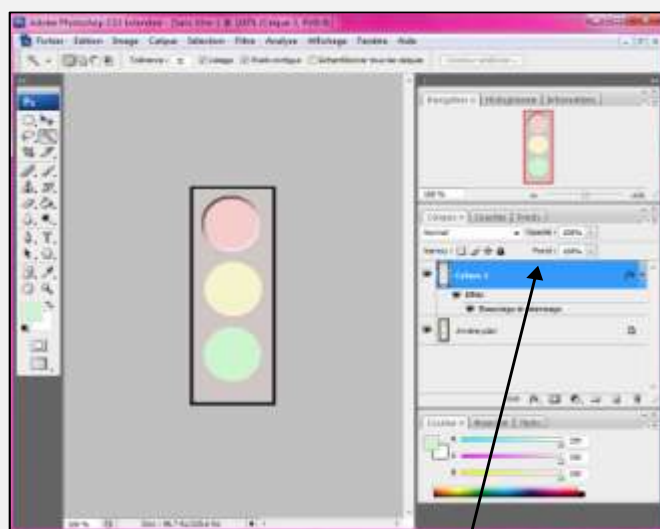
Étape 7 : Répéter la manœuvre précédente en dessinant un troisième cercle en dessous du précédent mais en appliquant cette fois ci une couleur **vert pâle**. À la fin, désélectionner en cliquant en dehors de la zone de travail ou en faisant **CTRL+D**



Étape 8 : A l'aide de l'outil **baguette magique**, sélectionner le premier cercle (rose pâle). Copier et coller le cercle sélectionné en faisant **CTRL+C** et ensuite **CTRL+V**. on peut également faire **Edition > Copier** et ensuite **Edition > Coller**. Un nouveau calque se crée automatiquement contenant la copie du rond. Sélectionner ce calque et faire **Calque > Style de calque > Biseautage et Estampage**. Valider avec les paramètres par défaut.



Fenêtre style de calque



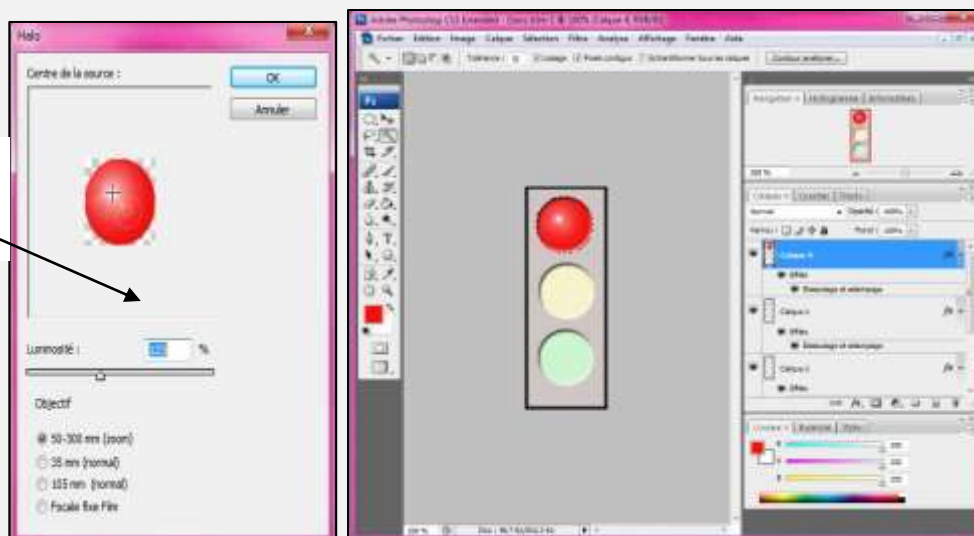
Nouveau calque créé automatiquement

Étape 9 : Se placer sur le calque d'arrière plan et répéter les tâches de l'étape 8 mais cette fois ci avec le second cercle (jaune pâle). Refaire la même opération avec le troisième rond (vert pâle).

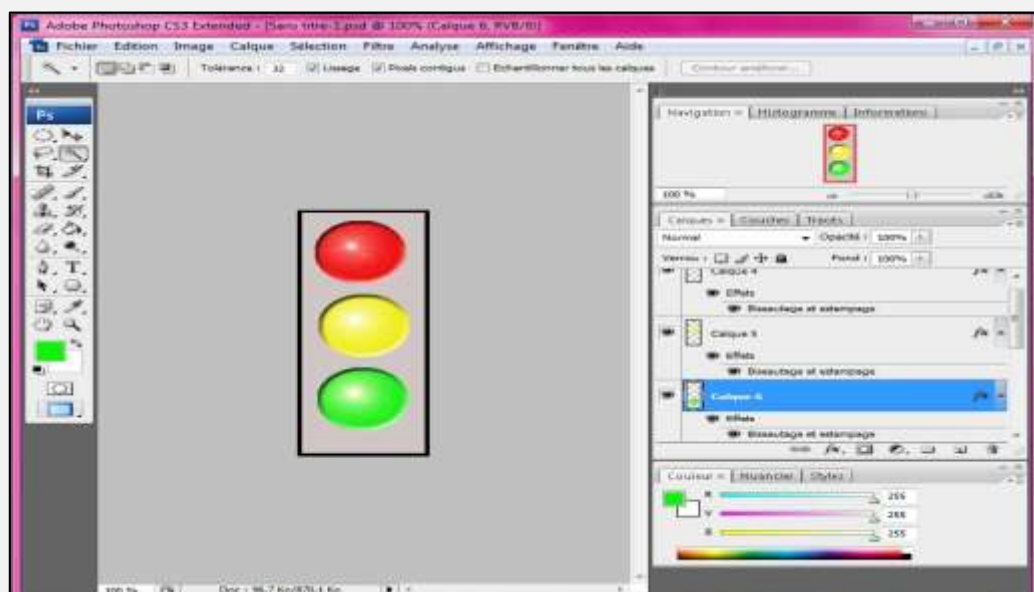
Étape 10 : Se placer sur le calque d'arrière plan et sélectionner le premier cercle à l'aide de l'outil **Baguette magique**. Créer un nouveau calque en suivant **Calque > Nouveau > Calque**. Avec un **cliquer – glisser**, placer le nouveau calque au dessus des autres en premier plan.

S'assurer qu'on est sur le nouveau calque créé, sélectionner une couleur **rouge vive** comme couleur de premier plan et appliquer la au cercle. Ajouter des effets au calque en exécutant la série d'opérations suivantes **Calque > Style de calque > Biseautage et Estampage** et ensuite **Filtre > Rendu > Halo** ; dans la fenêtre qui s'ouvre mettre la **luminosité à 125** et valider.

Fenêtre
Halo

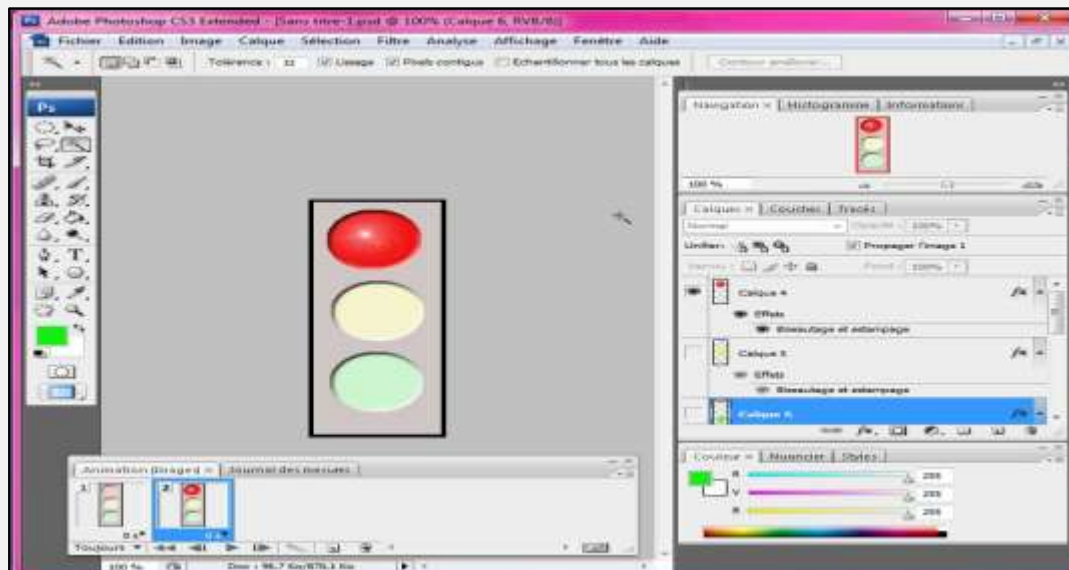


Étape 11 : Reprendre l'étape précédente mais avec le deuxième cercle, une **couleur jaune vive** et en plaçant le **nouveau calque après celui créé à l'étape 10**. Ré-exécuter les mêmes tâches avec le troisième rond, la **couleur verte vive** et en plaçant le calque créé **après les deux calques ci-dessus**.

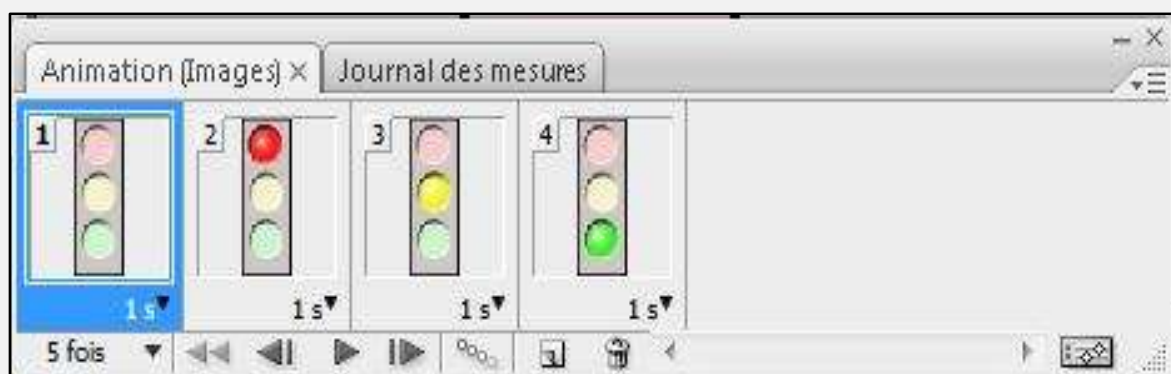


Étape 12 : Rendre **invisible les trois derniers calques créés** en cliquant sur l'icône  dans la palette de calque. Ouvrez **la palette d'animation**. Il s'ouvre avec comme première image l'image de la zone de travail (trois cercles de couleur pâle).

Étape 13 : **Dupliquer** cette première image. Se placer sur cette nouvelle image et rendre visible le calque contenant le premier cercle rouge vif.



Étape 14 : Dupliquer à nouveau la deuxième image insérée ci dessus. Se placer sur cette nouvelle image, rendre visible le calque contenant le deuxième cercle jaune vif et invisible le calque du cercle rouge vif. Refaire la même opération en dupliquant la dernière image insérée, avec le cercle vert vif visible et les autres invisibles.



Étape 15 : définir **le nombre de boucle à 5** et le **délai d'apparition de chaque image à 1 seconde**. Tester l'animation en cliquant sur le **bouton de lecture**.

Étape 16 : Enregistrer le document tout d'abord comme un projet modifiable avec l'extension **.PSD ou .PDD** ; pour enregistrer l'animation, faire **Fichier > Enregistrer pour le web et ses périphériques**. Dans la fenêtre qui s'ouvre, s'assurer que l'extension est **.gif** et valider.



Compléments

Certains logiciels offrent la possibilité de faire des animations ou encore des montages avec plusieurs photos numériques.



Résumé

Une **animation** est une séquence d'images affichées dans le temps. Chaque image diffère très légèrement de la précédente, créant une illusion de mouvement lorsque les images sont visualisées en succession rapide. L'animation se fait à partir des palettes animation et calque.

La palette animation offre diverses opérations parmi lesquelles :

- ⓐ Définition de la boucle et du délai d'apparition des images ;
- ⓐ La sélection, réorganisation, ajout, suppression d'images

La palette des calques facilite la manipulation des calques. Les calques sont des couches transparentes superposées. L'image est déposée sur un calque. Ces derniers permettent de travailler sur un objet sans altérer les autres.

Le montage d'une animation nécessite une bonne conception au départ.



EXERCICE 1 :

1. Présenter le concept d'animation
2. Quelles sont les palettes qui permettent de faire une animation ?
3. Citer 05 boutons de la palette d'animation

EXERCICE 2 :

1. Donner deux opérations qu'on peut effectuer dans la palette d'animation
2. Donner les étapes de réalisation d'une des opérations ci-dessus.
3. A quoi sert la palette des calques ?
4. Définir et donner le rôle d'un calque.

EXERCICE 3 :

1. Écrire le mot suivant PROBATOIRE dans la zone de travail en plaçant chaque lettre du mot dans un calque différent.
2. Appliquer une couleur différente à chaque lettre du mot.
3. Appliquer des effets à chaque calque contenant une lettre.
4. Ouvrir la palette d'animation, insérer les différentes et monter une animation faisant apparaître de façon progressive chaque lettre du mot.
5. Enregistrer le document comme un projet et ensuite comme un fichier inséré dans une page web.

MODULE

2

ALGORITHMIQUE

- 1- Notion d'algorithme*
- 2- Instructions simples,
procédures et tableaux*
- 3- Les structures
algorithmiques*
- 4- Exécution d'un algorithme*

Chapitre 5

NOTION D'ALGORITHMIQUE



SOMMAIRE



- I- ÉTAPES DE RÉOLUTION D'UN PROBLÈME
- II- ÉLÉMENTS DE LA STRUCTURE D'UN ALGORITHME
- III- STRUCTURE GÉNÉRALE D'UN ALGORITHME



Problème

Il vous est demandé de décrire de manière ordonnée l'ensemble des tâches dans qu'un élève effectue chaque matin de son levée du lit jusqu'à son arrivée à l'école.

- ⊕ Êtes-vous capable de lister toutes ces tâches ?
- ⊕ Quelles sont d'après vous les tâches qui renferment ou englobent d'autres sous tâches ?



Pré-requis

- Être capable de réaliser une étude de texte;
- Savoir organiser une situation en plusieurs étapes;



Objectifs

- Pouvoir définir un algorithme;
- connaître les éléments constitutifs d'un algorithme;
- Identifier les étapes de résolution d'un problème;
- Identifier les données, les traitements et les résultats d'un problème.

INTRODUCTION

Le mot « **algorithme** » provient de la forme latine (**Algorismus**) du nom du mathématicien arabe **AL-KHAREZMI** ou **AL-KHW - ARIZM** (né vers 780 - mort vers 850) qui a écrit en langue arabe le plus ancien traité d'algèbre « abrégé de calcul par la complétion et la simplification » dans lequel il décrivait des procédés de calcul à suivre étape par étape pour résoudre des problèmes ramenés à des équations.

Un algorithme est une suite finie et ordonnée d'instructions permettant de résoudre des classes de problèmes semblables en un temps fini. Il est également perçu comme un ensemble de règles opératoires rigoureuses permettant d'exécuter dans un ordre déterminé un nombre fini d'opérations élémentaires selon une syntaxe bien définie. Cette syntaxe est définie par le langage algorithmique qui est plus proche du langage naturel (humain).

L'algorithmique est la science qui étudie les algorithmes

On trouve des algorithmes dans des situations de la vie courante (cuisiner, se vêtir, construire une maison, traverser la chaussée d'une autoroute, préparer un plat de éru, s'inscrire à l'examen du probatoire, mettre en marche un ordinateur, téléphoner, la conduite d'un train, la consultation d'un catalogue de bibliothèque, etc.). Bref, tout ce qui se passe dans la vie se fait suivant un algorithme.

Les propriétés d'un bon programme

Les propriétés d'un bon programme sont :

- **La lisibilité** : les instructions doivent être bien édentées, commentées ;
- **L'exactitude** : l'aptitude d'un programme à fournir le résultat voulu et à répondre ainsi aux spécifications ;
- **La robustesse** : l'aptitude à bien réagir lorsque l'on s'écarte des conditions normales d'utilisation ;
- **L'extensibilité** : la facilité avec laquelle un programme pourra être adapté pour répondre à l'évolution des spécifications ;
- **La réutilisabilité** : la possibilité d'utiliser certaines parties du logiciel pour résoudre un autre problème ;
- **La portabilité** : la facilité avec laquelle on peut exploiter un même logiciel dans différentes implémentations ;
- **L'efficacité** : le temps d'exécution, taille mémoire.

I. ÉTAPES DE RÉOLUTION D'UN PROBLÈME

La résolution d'un problème peut se faire de plusieurs façons. L'utilisation des algorithmes pour résoudre un problème passe par étude préalable du problème structuré en trois principales étapes que sont :

- ❖ L'identification des données
- ❖ L'identification des opérations ou traitements
- ❖ L'identification des résultats

1. Identification des données

Il s'agit de repérer les **données nécessaires** voire **indispensables** à la résolution d'un problème. Les données sont généralement identifiables dans un problème par des noms ou groupes nominaux.

Ces données sont des objets manipulés par le programme. Elles peuvent être variables ou constantes. Ces dernières sont de divers types : numérique (entier, réel, etc.), les textes (caractère, chaîne de caractères, etc.), logique (booléen), tableaux, etc.

2. Opérations de traitement

Les opérations de traitement sont des actions ou des tâches à effectuer. Il peut s'agir d'une instruction simple ou d'une succession d'instructions. Les opérations de traitement sont généralement identifiées par des verbes d'action. Il peut arriver que certaines opérations de traitement produisent des résultats intermédiaires qu'il est bon de mémoriser pour la suite de l'exécution du programme. Ces résultats intermédiaires deviennent des données (données intermédiaires) s'ils sont utilisés dans la suite du programme.

3. Identification des résultats

Les résultats sont obtenus à la suite de l'exécution d'une instruction ou d'une succession d'instructions. On peut avoir des résultats intermédiaires et des résultats finaux. Ces derniers sont obtenus à la fin de l'exécution du programme.



ACTIVITÉ : Activité préparation de la sauce de tomate au poulet

Votre sœur veut préparer un plat de sauce de tomate au poulet pour la première fois et la recette qui lui est proposée est la suivante :

Ingrédients : 1 poulet entier, 10 grosses tomates, 1 oignon, une gousse d'ail, 1 poivron vert, 3 cubes, 1 citron, 3 grosses carottes, 1 pincée de poivre blanc, poivre jaune, anis vert, thym, persil, céleri, Huile raffinée, 1 piment rouge.

Préparation :

- Déplumer et nettoyer le poulet
- Découpez le poulet en des petits morceaux et mettez-les dans une casserole.
- Émincez la moitié de l'oignon sur les morceaux de poulet.
- Ajoutez-y une pincée de sel, thym, ail, jus du citron et 2 verres d'eau.
- Couvrez la casserole et laissez macérer à feu doux pendant 20 mn puis retirer.
- Entre temps, émincez le reste d'oignon. Mixez tomates, piment et anis vert.
- Coupez les carottes en petits dés et hachez le poivron en fines lamelles.
- Mettez de l'huile dans une casserole et laissez chauffer environ 1 minute.
- Faites frire les morceaux de poulet, puis retirer les.
- Ajoutez ensuite dans la casserole les lamelles d'oignon et de poivron, les carottes en dés. Laissez mijoter pendant 3 minutes en remuant.
- Ajoutez la tomate mixée. Laissez à nouveau mijoter pendant 6 minutes.
- Incorporez à présent le poulet avec sa marinade. Rallongez avec de l'eau. Laissez mijoter pendant 20 mn.

I. Quelles sont les données que l'on retrouve en entrées de ce problème ?

Solution

Les données en entrée sont : 1 poulet entier, 10 grosses tomates, 1 oignon, une gousse d'ail, 1 poivron vert, 3 cubes Maggi, 1 citron, 3 grosses carottes, 1 pincée de poivre blanc, poivre jaune, d'anis vert, du thym, du persil, du céleri, de l'huile raffinée, 1 piment rouge, du sel, de l'eau, la casserole, le feu, le couteau, le mixeur, la louche, les spatules, les plats, les récipients, etc.

Application : Un élève veut préparer d'un plat d'omelette. Ecrire tout ce qu'il faut et donner les étapes de ladite préparation. Puis, ressortir les données que l'on retrouve en entrée pour ce problème.

2. Quelles sont les opérations de traitement ?

Solution

Les opérations de traitement que l'on retrouve dans ce problème sont :

Déplumer le poulet ; nettoyer le poulet ; Découper le poulet en morceaux assez fins ; mettre dans une casserole adaptée les morceaux de poulet ; Émincer la moitié de l'oignon sur les morceaux de poulet ; ajouter une pince de sel, thym, ail, jus du citron et 2 verres d'eau ; Couvrir la casserole ; Émincer le reste d'oignon ; Mixer les tomates, piment et anis vert.

Application : Énumérer la suite des traitements

3. Quels sont les résultats ?

Solution

Le résultat final est la sauce de tomate au poulet. Les résultats intermédiaires sont nombreux : on peut citer entre autre le poulet déplumé, les morceaux de poulet, les oignons émincés, etc.

Application :

- Pour chacune des instructions donner les résultats intermédiaires obtenus.
- Construire le schéma suivant pour chaque instruction :



Où le traitement représente une action, les résultats représentent les éléments obtenus à la fin de l'action et les données en entrée sont les objets sur lesquels on va agir.



COMPÉTENCES REQUISES POUR UNE BONNE DÉMARCHE ALGORITHMIQUE

La sensibilisation de l'élève à la question de la « démarche algorithmique » pourra se faire en évitant toute technicité ou exposé systématique. Les compétences suivantes pourront être identifiées et travaillées :

- ⊕ comprendre et analyser un algorithme préexistant ;
- ⊕ modifier un algorithme pour obtenir un résultat particulier ;
- ⊕ analyser la situation : identifier les données d'entrée, de sortie, le traitement...;
- ⊕ mettre au point une solution algorithmique : comment écrire un algorithme en « langage courant » en respectant un code, identifier les boucles, les tests, des opérations d'écriture, d'affichage... ;
- ⊕ valider la solution algorithmique par des traces d'exécution et des jeux d'essais simples.

II. ÉLÉMENTS DE LA STRUCTURE D'UN ALGORITHME

1. Langage de définition d'algorithme (LDA)

Un **Langage de définition d'algorithme (LDA)** est un langage naturel qui utilise un ensemble de mots-clés et de structure permettant de décrire de manière complète, claire, l'ensemble des opérations à exécuter sur des données pour obtenir des résultats ; on n'hésitera donc pas à agrémenter un LDA de nombreux commentaires.

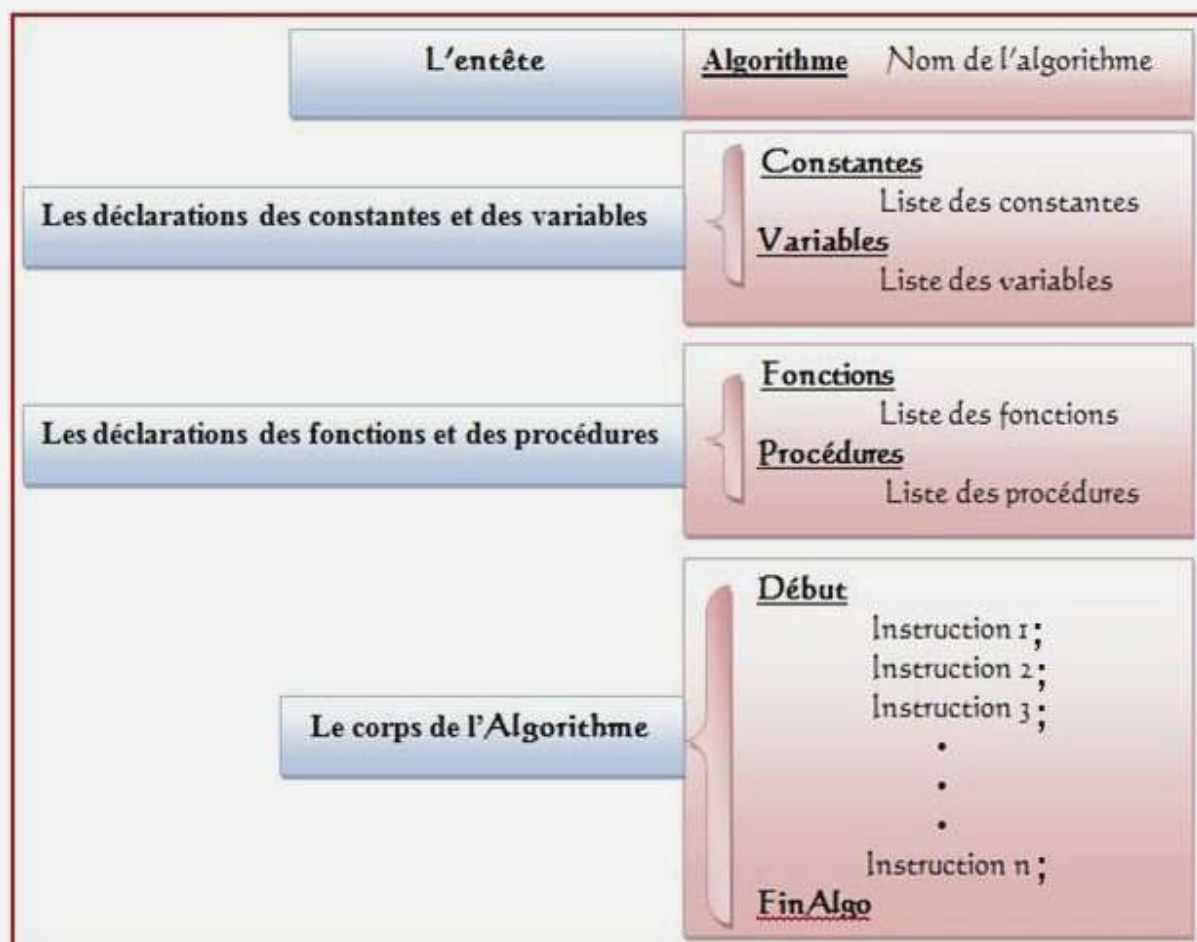


- Il n'existe pas un LDA universel permettant de définir n'importe quel problème faisant appel à un algorithme. Un LDA dépend fortement du problème posé.
- L'avantage d'un tel langage est de pouvoir être facilement transcrit dans un langage de programmation.

2. Structure d'un algorithme

a. Représentation

Un algorithme est structuré de la façon suivante :



- ⓐ **L'en-tête** permet tout simplement d'identifier un algorithme par un nom. Ce nom n'influence en rien le bon déroulement de l'algorithme. En général il faut donner des noms parlants à nos algorithmes, ceci pour permettre au lecteur d'avoir une idée de ce que fera l'algorithme;
- ⓐ **Les déclarations de constantes, variables, structures** sont une liste exhaustive des objets ou des données utilisés et manipulés dans le corps de l'algorithme.
- ⓐ **Les déclarations des fonctions et des procédures** sont une liste exhaustive des opérations non élémentaires c'est à dire qui contiennent d'autres instructions simples ou non. Les procédures et fonctions seront appelées dans le corps de l'algorithme ;
- ⓐ **Le corps de l'algorithme** contient les tâches (instructions, opérations) à exécuter. Ces tâches peuvent être des appels de fonction ou des instructions simples

b. Les mots clé

Dans un LDA, certains mots sont réservés pour un usage bien défini. On les nomme les **mots clés**. Ce sont les mots que le langage utilise pour son fonctionnement. Dans notre langage de définition les mots clés commenceront toujours par une majuscule et seront soulignés. Les mots clés dans la structure globale d'un algorithme ci-dessus sont donnés comme suit :

- ⓐ **Algorithme** : il précède le nom à l'algorithme ;
- ⓐ **Constantes** : ce mot clé de l'algorithme introduit la déclaration des constantes. Une constante est un objet dont le contenu reste invariant lors de l'exécution d'un algorithme.
- ⓐ **Variables** : ce mot clé introduit la déclaration des variables. Une variable est un objet dont le contenu peut changer au cours de l'exécution de l'algorithme. Une variable a un **type**.
- ⓐ **Fonction** ou **Fonc** : ce mot clé introduit la déclaration des fonctions à utiliser dans le corps de l'algorithme.
- ⓐ **Procédures** ou **Proc** : ce mot clé introduit la déclaration des procédures.
- ⓐ **Début** : il marque le commencement de l'algorithme ;
- ⓐ **FinAlgo** ou **Fin** : il marque la fin de l'algorithme ;

En plus de ces mots clés, on rencontre d'autres mots clés tels que :

- ⓐ **Réel, Caractères, Entiers, Chaîne, Booléen...** sont des mots clés qui permettent de définir des types (nous y reviendrons un peu plus loin dans cet ouvrage). Le **type** détermine la nature de la variable ou de la constante se trouvant dans un algorithme. Ces mots clés ne sont généralement pas soulignés ;
- ⓐ **Si... Finsi, Tantque ...Fintantque, Pour ...Finpour, Répéter ... Jusqu'à**, etc. sont des mots clés permettant de définir les structures itératives, conditionnelles, etc.

- © **Co...Fco** permet de définir un **commentaire** qui est un texte expliquant les éléments d'un algorithme et destiné à un quelconque lecteur de l'algorithme et qui n'a aucune incidence sur son exécution. Un commentaire peut s'étendre sur une ou plusieurs lignes.



Différences entre constantes et variables

Une constante est un objet dont le contenu reste invariant lors de l'exécution d'un algorithme. Dans tout le corps (traitement) de l'algorithme, on ne peut pas modifier un objet de ce type ni faire aucune opération sur celui-ci.

Une variable est un objet dont le contenu peut changer au cours de l'exécution de l'algorithme. Toute valeur qu'on affecte à une variable (donnée doit être du même type que celle-ci (la variable)).

La **déclaration d'une variable** ou d'une **constante** nécessite de préciser son type (**Réel, Caractères, Entiers, Chaîne, Booleen, tableau,...**).

Exemple : les éléments tels que poulet, carotte, oignon, tomate, poivron, eau, etc. sont des variables dans ce sens qu'ils sont modifiés au cours de la préparation. Tandis que les éléments tels que marmite, couteau, louche, etc. sont des constantes car ils ne sont pas modifiés au cours du processus de cuisson.

Différences entre procédures et fonctions

Une procédure est un ensemble d'instructions référencées par un nom et dont l'exécution est provoquée par la simple énonciation de ce nom. Une procédure ne renvoie pas un résultat.

Comme pour la procédure, l'exécution d'une fonction est provoquée par la simple évocation de son nom, à la différence qu'elle doit renvoyer une valeur de retour dont le type doit être défini.



Comment différencier une variable de sa valeur.

Une variable peut être vue comme un contenant et sa valeur le contenu. On peut voir une variable comme un tiroir comportant une étiquette (cette étiquette est le nom de la variable) et sa valeur comme le contenu du tiroir (les dossiers, les bijoux, etc.). La variable est du même type que son contenu c'est-à-dire, le contenu d'une variable définit son type. **Exemple** on peut avoir sur un tiroir dont l'étiquette est **mes bijoux de mariage**. Dans ce tiroir sont rangées l'alliance, la gourmette, les boucles d'oreilles, la chaîne etc. On peut qualifier ces éléments de **Bijoux**. On dira donc que la variable **mes bijoux de mariage** est de type **Bijoux** et peut prendre les valeurs alliance, gourmette, boucle d'oreilles, chaîne, etc. qui sont tous de type Bijoux.

c. Les identificateurs et les séparateurs

Les identificateurs

Un **identificateur** est une caractéristique de tout ce qui peut être déclaré.

Dans un programme, des éléments comme des variables, fonctions, types, ... sont désignés par le terme **d'identificateur**. Comme dans la plupart des langages, un tel identificateur est formé d'une suite de caractères choisis parmi les lettres, chiffres ou le caractère souligné (`_`), le premier d'entre eux étant nécessairement différent d'un chiffre.

Les majuscules et les minuscules sont autorisées mais ne sont pas équivalentes. Ainsi, les identificateurs **ligne** et **Ligne** désignent deux objets différents.

Voici quelques identificateurs corrects :

`lg_lig`, `valeur_5`, `_total`, `_89`, `la`, `La`, etc.

les séparateurs

Dans le langage naturel, les mots sont séparés par un espace, un signe de ponctuation ou une fin de ligne. Il en va de même pour notre langage dont les règles vont paraître naturelles.

Les caractères séparateurs comprennent tous les opérateurs (`+`, `-`, `*`, `=`, `==`, etc.) ainsi que les caractères dits de ponctuation : `(` `)` `[` `]` `,` `.` : etc.



Le point-virgule (`;`) marque la fin d'une instruction



ACTIVITÉ : Écrire un algorithme qui permet d'afficher MERCI à l'écran

Entête : (nom de l'algorithme)

Algorithme afficher /* cet algorithme permet d'afficher MERCI à l'écran */

Partie déclaration

Variables :

a de type ChaîneCaractère /* a est une variable de type chaîne de caractère */

Corps de l'algorithme

Début

a « MERCI » ; /* on affecte la chaîne de caractère MERCI à la variable a */
Afficher(a) ; /* permet d'afficher le contenu de a à savoir MERCI à l'écran */

FinAlgo

APPLICATION : Écrire un algorithme qui permet de saisir sur le clavier, d'afficher à l'écran et d'effacer le mot PREMIÈRE C/D à l'écran.



Différence entre un algorithme et un programme

Un programme est l'écriture d'un algorithme dans un langage de programmation donné (Pascal, C, C#, C++, Java, etc.). Un langage de programmation est un ensemble de règles de vocabulaire et de grammaire compréhensible par un ordinateur. Les phrases d'un langage sont des **instructions**. Une instruction est un ordre que l'on donne à la machine. Ces langages de programmation diffèrent du langage algorithmique en ce sens qu'ils sont compréhensibles par un ordinateur ; le langage algorithmique quant à lui est plus proche du langage humain qui n'est pas compréhensible par un ordinateur.



Résumé

Une **donnée** est une information élémentaire non décomposable;

Une donnée a un **type** qui peut être **entier**, **caractères**, **chaîne de caractère**...

Pour repérer les données dans un texte, on peut :

- ⊗ faire *l'inventaire* de tout les noms et groupes nominaux implicites et explicites
- ⊗ *faire une catégorisation dans la famille de mots*, car il ya des mots qui ont la même signification.

Une **opération de traitement** est une action effectuée par les acteurs du système.

Pour repérer les opérations de traitement, *il faut recenser les verbes d'actions*.

Les **données en sortie** sont des données résultant d'une opération.

L'écriture d'un algorithme se fait à l'aide d'un *langage de définition d'algorithme (LDA)*. Ce LDA fait appel à de nombreuses notions telles que les mots, les commentaires, les identificateurs...

Un identificateur est une suite de lettre et de chiffre différents des mots clés et ne pouvant pas commencer par un chiffre.

Les **mots clés** sont un ensemble de mots **prédéfinis**. Ce sont les mots que le langage utilise pour son fonctionnement et ne peuvent par conséquent pas être utilisées comme identificateurs.

Pour une bonne compréhension du langage il est très important d'utiliser les **commentaires**.

Un algorithme a trois grandes parties : **L'entête, la partie déclarative et le corps de l'algorithme.**



Exercice 1 :

- 1) Qu'est ce qu'un algorithme, l'algorithmique, une donnée?
- 2) Quelles sont les propriétés d'un bon algorithme ?
- 3) Expliquez succinctement les différentes étapes de résolution d'un algorithme.
- 4) quelles sont les aptitudes nécessaires à la production d'un bon algorithme ?

Exercice 2 :

- 1) Quels sont les principaux mots clés de notre langage algorithme ?
- 2) Quels sont les symboles qui permettent de caractériser un commentaire ?
- 3) Qu'est ce qu'un identificateur ?
- 4) Citez quelques séparateurs.
- 5) Donnez la structure générale d'un langage.

Exercice 3:

Parmi les objets suivants, dites lesquels peuvent être considérés comme variables, comme constantes : Un sac, Une marmite, de l'eau, un cahier, du sucre, des beignets, une bouteille, une assiette, une pièce d'argent, une maison, la télé, la boîte académique.

Exercice 4:

Soit la suite d'instruction suivante : se lever ; se brosser les dents ; se laver ; prendre le chemin de l'école. Écrire un algorithme nommé « aller à l'école » qui donne la suite d'instruction à suivre pour aller à l'école. Cet algorithme n'aura pas de partie déclaration.

Exercice 5:

Soit à considérer les objets suivants : Un verre, un sachet, de l'eau, du lait, du sucre. On désire écrire un algorithme qui permet de faire une tasse de lait pour le petit déjeuner.

- i. Parmi les objets cités ci-dessus, dites ceux qui sont des variables et ceux qui sont des constantes.

On suppose que vous avez les instructions suivantes : tourner, bouillir de l'eau, verser de l'eau dans le verre, verser du lait dans le verre, déchirer le sachet de lait, verser du sucre dans le verre.

- ii. Mettez ces instructions dans un ordre cohérent afin d'écrire un algorithme nommé préparation du lait qui permet de faire une tasse de lait.

Exercice 6 :

Écrire un algorithme portant sur :

- i. La mise en marche d'un ordinateur.
- ii. L'achat d'un billet dans une agence de voyage.
- iii. La recherche du numéro de téléphone d'un ami dans le téléphone.

Exercice 7 :

Mme EWEIH veut préparer du plantain pour consommer avec le ndolè qu'elle a déjà apprêtée. Pour cela elle suit les étapes suivantes :

Elle commence par éplucher le plantain qu'elle met dans une bassine. Ensuite elle y verse de l'eau avec un gobelet jusqu'à ce que la quantité soit suffisante pour le lavage. Après avoir fini de laver le plantain, elle le verse dans la marmite, elle y met à nouveau de l'eau, un peu de sel, et dépose la marmite au feu. Lorsque le plantain est cuit, elle éteint le feu.

- 1) Recensez les objets à caractère variable de ce problème.
- 2) Recensez les objets à caractère constant du problème.
- 3) Donnez les principales étapes à suivre pour la préparation du manioc.
- 4) On suppose que ces étapes sont des instructions à suivre pour la préparation du manioc. Écrire un algorithme qui permet de préparer le manioc.

Exercice 8 :

Un poste de distribution automatique de carburant d'une station service dispose de trois pistolets (un pour le gasoil, un pour l'essence sans plomb, un pour l'essence super) et de trois compteurs (« prix unitaire », « volume », « prix à payer »). Le compteur « prix à payer » n'est pas programmable par le client.

L'un des compteurs « volume » et « prix à payer » est programmé par le client pour indiquer soit la quantité de carburant qu'il désire. Chaque poste est soit disponible, soit en cours d'utilisation, soit en panne. On ne s'intéresse pas au stockage de carburant. Un événement « rupture de stock » rend indisponible des postes concernés. Chaque client qui effectue une transaction normale est supposé être servi de la quantité de carburant qu'il a demandé si sa carte d'abonné le lui permet.

Les postes disponibles fonctionnent de la façon suivante : l'abonné insère sa carte dans le lecteur du poste de distribution automatique, sélectionne le type de carburant, et indique la quantité voulue. La transaction est initialisée par la mise à zéro du compteur non programmé. Le client tire le pistolet correspondant à son carburant, le moteur se met en route et propulse le carburant demandé lorsque le client appuie sur la gâchette du pistolet. Le compteur non programmé par le client évolue, selon un pas fixe à l'avance lors de l'installation du poste de distribution, avec la propulsion du carburant. Lorsque le client relâche la gâchette, la propulsion du liquide est stoppée. Cette propulsion peut se faire en plusieurs fois. Lorsque le client remet le pistolet dans son emplacement, la phase de service est terminée.

1. Quelles sont les données que l'on retrouve en entrées de ce problème ?
2. Donner quelques opérations de traitement que l'on retrouve dans ce problème ?
3. Quelles informations peuvent-être considérées comme étant un résultat ou une sortie de ce problème ?

Chapitre 6

INSTRUCTIONS SIMPLES, PROCÉDURES ET TABLEAUX



SOMMAIRE



- I. OPERATIONS SUR LES VARIABLES
- II. LES TABLEAUX
- III. PROCÉDURES ET FONCTIONS



Problème



Pré-requis

- Identifier les étapes de résolution d'un problème
- Identifier les données, les traitements et les résultats d'un problème.



Objectifs

- Savoir déclarer, affecter et incrémenter une variable
- Lire et écrire une information
- Savoir déclarer et appeler une fonction ou une procédure
- Connaître la notion de récursivité
- Connaître les notions de base des tableaux

INTRODUCTION

Certains de nos problèmes demandent des efforts énormes aussi bien cognitifs que psychomoteurs. Alors, pour alléger ce problème, l'Homme a souvent recourt à la machine afin de lui faciliter la tâche. Mais pour y arriver nous devons nous poser un certains nombres de questions d'où la notion **d'algorithme**.

Une fois les étapes de résolution d'un problème réalisées, on peut maintenant résoudre le problème en écrivant l'algorithme correspondant. Un ensemble d'éléments sont très fréquent dans l'écriture de ces algorithmes à l'instar du nom de l'algorithme, des variables, de même qu'un ensemble d'opérations telles que l'affectation, l'incrémentation, la lecture à l'écran, l'écriture à l'écran, les fonctions, la notion de récursivité des fonctions, les tableaux et bien d'autres.

Il sera question pour nous dans la suite de ce chapitre de présenter de façon simple tous ces éléments et opérations suivis d'exemples.

I. OPÉRATION SUR LES VARIABLES

Les variables sont des éléments indispensables dans la réalisation d'un algorithme. Lors de la cuisson d'une omelette, on a besoin d'un plat pour les oignons et tomates un autre dans lequel on va battre l'œuf ; ces différents plats sont vue comme des variables.

1. La déclaration des variables

La déclaration : c'est une liste exhaustive des objets, grandeurs utilisés et manipulés dans le corps de l'algorithme. Cette liste est placée en début d'algorithme juste après le nom de l'algorithme.

Variables : c'est une partie de l'algorithme qui permet de déclarer des variables ; Une variable a un **type** et tout objet qu'on affectera à une variable d'un type donné doit être du même type que la variable. En d'autres termes, une variable peut être vue comme un contenant dans lequel on y met des contenus. Les types représentent le type de contenu que peut prendre le contenant.

Une variable : c'est un objet dont le contenu peut changer au cours de l'exécution de l'algorithme. Elle permet d'identifier une donnée sur laquelle agit un programme. Les noms de variables doivent être choisis avec soin. Par exemple $\text{perimetreCarre} = \text{coteCarre} \times 4$.

Le mot clé permettant d'effectuer la déclaration d'une variable est **Var**. la syntaxe est la suivante : **Var Nomdevariable : Type de la variable ;**

Exemple 1 :

*Var nombre1 : entier ; var nombre2 : réel ; var lettre1 : caractère ;
Var trouver : Booléen ; var rayon : réel ; var nom : chaîne de caractères ;*

NB : Le mot clé *Var* n'apparaît qu'une seule fois dans un algorithme.

Exemple 2 :

Supposons que l'on veuille mettre de l'eau et de l'huile dans deux bidons différents. Ces deux bidons en algorithmique, bien qu'étant identiques physiquement sont différents. Un bidon d'eau ne saurait remplacer un bidon d'huile. Le premier est une variable de type « bidon d'eau », le second une variable de type « bidon d'huile ».

Plusieurs opérations peuvent être effectuées sur des variables. On peut remplacer, ajouter ou diminuer le contenu d'une variable.

2. La Constante

Partie de l'algorithme qui permet de déclarer des constantes.

Une **constante** est un objet dont le contenu reste invariant lors de l'exécution d'un algorithme.

Le mot clé est **const** ou *constante*.

La syntaxe est la suivante: **const NomConstante = Valeur ;**

Exemple : *const pi = 3.14 ; const N = 100*



Contrairement aux variables, on ne peut pas modifier un objet de type constante ni faire aucune opération sur celui-ci.

3. L'Affectation

L'**Affectation** est une opération qui consiste à donner une valeur à une variable. Une valeur peut être affectée directement à cette variable ou par l'intermédiaire d'une autre variable contenant ladite valeur. Affecter une valeur à une variable, c'est remplir une case mémoire : toute variable doit être initialisée avant d'être utilisée.

Le signe **=** peut être utilisé pour faire des affectations de même que le symbole **<-** ou à l'aide du mot clé **prend la valeur de**.

L'affectation est un **opérateur binaire** c'est-à-dire qu'elle ne fait intervenir que deux choses, ceci de part et d'autre du symbole.

Sa syntaxe est la suivante :

Variable <- expression qui se lit « **Variable prend la valeur de expression** ».

Après l'affectation, le contenu de « **Variable** » est modifié. Il contient désormais, la valeur de **l'expression** de droite. Son mode de fonctionnement est le suivant : On évalue d'abord l'expression de droite avant de l'affecter à « Variable »



- Ⓢ la variable doit être de même type que l'expression. Son type est celui avec lequel il a été défini ;
- Ⓢ l'affectation n'est pas commutative c'est-à-dire que $a=b$ est différent de $b=a$;
- Ⓢ elle est instantanée et valable à un moment donné ;



Voici un algorithme qui est écrit de trois manières différentes. Dire ce que fait cet algorithme. En supposant que x vaut 6, que valent u et y à la fin de cet algorithme ? Refaire la même opération lorsque x vaut 8, puis 12.

Entrée
Saisir x
Début
 u prend la valeur $x+4$
 y prend la valeur $u * x$
Afficher y
Fin

Entrée
Saisir x
Début
 $u = x+4$
 $Y = u * x$
Afficher y
Fin

Entrée
Saisir x
Début
 $u \leftarrow x+4$
 $Y \leftarrow u * x$
Afficher y
Fin

4. L'incrémentation

L'incrémentation est opération qui consiste à augmenter d'une unité le contenu d'une variable. Cette opération peut se faire de trois façons différentes. Soit x la variable à incrémenter :

- Ⓢ $x = x + 1$; ici on ajoute une unité de façon explicite
- Ⓢ $x++$; ici on ajoute une unité de façon implicite de même que $++x$

Ces deux façons de faire sont équivalentes.

On peut aussi vouloir diminuer une variable d'une unité c'est la **décrémentation**. Soit y la variable à décrémenter :

- Ⓢ $y = y - 1$; ici on diminue une unité de façon explicite
- Ⓢ $y--$; ici on diminue une unité de façon implicite de même que $--y$

Ces deux façons de faire sont équivalentes



Soit les expressions suivantes. Var z, y : entier $x=4$; $y=8$;

Soit la suite d'opérations suivante : $x = x - 1$; $y = y + 1$; $x--$; $y--$; $x = y + 2$; $y++$; $++x$;

- Ⓢ Quels sont les valeurs de x et y à la fin de ces différentes opérations ?
- Ⓢ Refaire les mêmes opérations pour $x=-3$ et $y=2$; pour $x=-5$ et $y=-1$; pour $x=6$ et $y=-12$.

5. Écriture et lecture à l'écran

Ⓢ Écriture à l'écran

Cette opération consiste à afficher ou écrire quelque chose à l'écran. Le mot clé utilisé par cette opération est **afficher**. La syntaxe est la suivante: **afficher ('je suis en classe de première scientifique')** ; « je suis en classe de première scientifique » sera affiché tel quel à l'écran.

Exemple : afficher ('veuillez entrer votre classe') l'expression qui apparaîtra à l'écran sera : veuillez entrer votre classe.

Ⓢ Lecture à l'écran

Cette instruction permet d'entrer une donnée à partir du clavier. La machine attend que l'utilisateur tape une valeur au clavier. Le mot clé ici est **saisir**. La Syntaxe est la suivante: **saisir (nom_variable)** nom_variable étant une variable d'un type déclaré à l'avance.

Exemple : saisir $var1$; $var1$ étant une variable d'un type déclaré au préalable.

En combinant les 2 instructions précédentes, nous avons :

afficher ('veuillez entrer votre classe')

Saisir (var1)



Ici, **vari** doit être une variable de type chaîne de caractère. Lorsque l'utilisateur entre sa classe, la machine récupère ce nom de classe et l'affecte à la variable **vari**. En affectant *Première C* à la variable **classe**; ce qui s'affiche à l'écran est :

Veuillez entrer votre classe
André

II. TABLEAUX

Le **tableau** est une structure permettant d'accéder de façon indépendante à un ensemble de variables de même type.

Il peut être également défini comme une structure de données permettant d'effectuer un même traitement sur des données de même nature.

Exemple :

5	-3	7	87	-54	10
---	----	---	----	-----	----

Ce tableau contient six éléments de même type : le type **entier**.

Avant toutes choses sur un tableau, il faut d'abord le créer. Pour créer un tableau il faut donner d'abord le type de données qu'il va contenir, ensuite son nom (ce nom doit être parlant) et enfin le nombre d'éléments qu'il peut contenir (taille). Sa syntaxe est la suivante :

Type nom_du_tableau [taille]

Exemple :

	1	2	3	4	5	6	7
TabNotes	12	05	07	18	15	09	16

Indice du tableau (pointe vers les indices 1 à 7)

Nom du tableau (pointe vers TabNotes)

Contenu du tableau (pointe vers les valeurs 12, 05, 07, 18, 15, 09, 16)

Si par exemple on veut créer un tableau de taille 50 renfermant les moyennes des élèves on procèdera comme suit :

Réel *tabMoyenne[50]*. Où *tabMoyenne* est le nom du tableau de moyenne.

Si on a une variable de type réel déclaré comme suit **réel x** on peut réaliser les opérations suivantes : **x = tabMoyenne[5]** qui n'est rien d'autre que l'affectation du **5ième** élément du tableau *tabMoyenne* dans la variable **x**. l'opération **tabMoyenne[3] = 28** est la modification du **3ième** élément du tableau *tabMoyenne* par la **valeur 28**.

On peut effectuer des opérations avec les éléments d'un tableau en modifiant directement leurs contenues. En supposant que la moyenne de la classe numéros 27 a une erreur d'une demi-moyenne (0.5) et qu'il faille arranger cela. Pour ce faire, on va

directement modifier cela sur la tableau ceci de la manière suivante : `tabMoyenne[27] = tabMoyenne[27] + 0.5 ;`



- ⊕ Un tableau est de **taille fixe**, ses données son de **même type** et sont **contiguës** en mémoire.
- ⊕ le premier élément d'un tableau est le plus souvent numéroté 0 en informatique. Ainsi un tableau de 8 éléments est numéroté de 0 à 7.
- ⊕ Mais dans notre cas le premier élément du tableau sera numéroté 1.
- ⊕ le nom du tableau doit être en un seul mot. S'il doit être constitué de plus d'un mot, il faudra les relier à l'aide de la barre de 8.
- ⊕ La taille du tableau doit être **un entier naturel**

Traitements opérant sur des tableaux

Sur un tableau plusieurs autres opérations peuvent être faites tel que :

- ⊕ créer des tableaux
- ⊕ ranger des valeurs dans un tableau
- ⊕ récupérer, consulter des valeurs rangées dans un tableau
- ⊕ rechercher si une valeur est dans un tableau
- ⊕ mettre à jour des valeurs dans un tableau
- ⊕ modifier la façon dont les valeurs sont rangées dans un tableau (par exemple : les trier de différentes manières)
- ⊕ effectuer des opérations entre tableau (addition, soustraction, fusion, comparaison, etc.)

Pour effectuer les opérations d'addition et de soustraction entre les tableaux, il faut qu'ils aient la même taille et les éléments de types compatibles.

III. LES PROCÉDURES ET FONCTIONS

1. Procédures

Une **procédure** est un sous-algorithme assurant de manière autonome un traitement particulier. Ce traitement peut alors être répété dans l'algorithme principal ou dans un autre sous-algorithme par simple appel de la procédure. La notion de procédure comporte deux (2) aspects :

- ⊙ Déclaration de la procédure : c'est la définition de la procédure ;
- ⊙ Appel de la procédure : c'est l'utilisation de la procédure.

NB :

- ⊕ La plupart des procédures informatiques stockent leurs opérations dans les bibliothèques ;
- ⊕ Une grosse procédure peut être décomposée en plusieurs petites procédures.

La structure générale d'une procédure n'est pas très différentes de celle d'un algorithme normal. Elle est semblable à celle d'un algorithme c'est-à-dire qu'elle comporte une **tête**, une partie **déclaration** et un **corps**.

La **tête de la procédure** commence par le mot réservé **PROCEDURE** suivi du nom de la procédure et d'une liste de paramètres placée entre parenthèses. Les paramètres assurent la communication entre les différents modules d'un algorithme.

Sa structure est la suivante :

Procédure nom de la procédure (liste des paramètres)
Déclaration des variables et constantes
Début
*/*Corps de la procédure*/*
Fin procédure

a. Paramètre d'une procédure

Un **paramètre** est une variable d'entrée de la procédure. Ces paramètres vont permettre l'appel de cette procédure dans une procédure principale ou appelante. La déclaration de ces variables se fait tel que présentée dans le chapitre 5.



- Ⓢ Pour une procédure ayant plusieurs paramètres, ceux-ci sont séparés par des virgules.
- Ⓢ Une procédure peut ne pas avoir de paramètres.

b. Déclaration et appel d'une procédure

La déclaration d'une procédure se fait de la façon suivante :

Procédure nom de la procédure (liste des paramètres)

Tout comme les programmes, le nom de la procédure n'est qu'indicatif mais il est prudent d'utiliser les noms parlants.

Exemple de déclaration d'une procédure : *Procédure addition (x : réel, y : réel)*

Cette procédure s'appelle addition et prend deux paramètres x et y qui sont des variables de type réels. **Lors de l'appel d'une procédure** les paramètres prennent le nom d'arguments un certain nombre de règles doivent être respectées.

- Ⓢ Le nombre d'argument doit être le même que le nombre de paramètre
- Ⓢ Les arguments doivent apparaître dans le même ordre des paramètres qu'ils représentent.
- Ⓢ Les arguments doivent être du même type que les paramètres qu'ils représentent.

2. fonction

a. La déclaration de la fonction

Une fonction est un sous-algorithme similaire à la procédure mais qui calcule une valeur d'un type donné. Cette valeur sera retournée à l'algorithme appelant à travers le nom de la

fonction. Le nom de la fonction est donc un paramètre résultat. Comme vous l'avez compris, la fonction diffère de la procédure par la valeur qu'elle **retourne** et son **type de retour**. De même que la procédure une fonction possède des paramètres. Sa structure générale est la même que celle d'une procédure. La déclaration se fait comme suit :

Fonction nom de la procédure (liste des paramètres) : Type de retour

Le type de retour est choisi parmi les types de bases ou construits tels que vu au chapitre 5.

Exemple : *Fonction addition (x : réel, y : réel) : réel* Cette fonction fait l'addition de deux nombre et le résultat retourné (type de retour) est un réel.

b. L'appel d'une fonction

L'**Appel d'une fonction** se fait comme celui de la procédure à la seule différence que le nom d'une fonction peut apparaître dans le membre droit d'une affectation.

Exemple d'appel de fonction :

Algorithme toto

...

...

Variable s : Réel

Constantes : ...

Début

...

S <- Addition (2, 4) ; / cette fonction additionne 2 et 4 et met le résultat dans s */*

...

Fin

3. La récursivité

Une **définition récursive** est une définition dans laquelle intervient ce que l'on veut définir. Un **algorithme** est dit récursif lorsqu'il est défini en fonction de lui-même. Ou encore on dit d'un algorithme qu'il est récursif lorsque lors de son déroulement il est réappelé sous certaines conditions.

On distingue plusieurs types de récursivités :

- ⊙ la récursivité simple ;
- ⊙ la récursivité multiple : un algorithme a une récursivité multiple lorsqu'il est défini en fonction de lui-même plusieurs fois ;
- ⊙ la récursivité mutuelle : des algorithmes sont dit mutuellement récursifs si elles dépendent les uns des autres ;
- ⊙ La récursivité imbriquée ;
- ⊙ la récursivité terminale : Un algorithme est dit récursif terminal s'il ne contient aucun traitement après un appel récursif ;



Compléments

- Ⓢ Les algorithmes récursifs et les fonctions récursives sont fondamentaux en informatique.
- Ⓢ Notion de **Dérécursivation** :
Dérécursiver, c'est transformer un algorithme récursif en un algorithme équivalent ne contenant pas d'appels récursifs.



Résumé

Lors de la résolution d'un problème écrit en langage algorithmique, plusieurs étapes sont traversées. Si l'algorithme est une fonction, on s'attendra à une valeur de retour ; par contre, si c'est une procédure on n'obtiendra rien en retour. On peut avoir ou non des paramètres chacune ayant un type.

Une fonction ou procédure peut avoir autant de paramètre voulu à condition de respecter leurs nombre et ordre lors de leur appel. A l'intérieur d'une fonction ou procédure, on a souvent un ensemble de variables et/ou constantes. Lors de la résolution du problème on peut se trouver en train de faire des affectations, des incrémentations, utilisation des tableaux, l'appel d'autre fonction et/ou procédure. Cette fonction ou procédure peu être récursif.



Exercice 1 : Questions de cours

- 1) Combien de signe d'affectation connaissez-vous ? les énumérer ?
- 2) Donner deux façons de faire une incrémentation
- 3) Donner deux façons de faire une décrémentation
- 4) C'est quoi une procédure ? c'est quoi une fonction ?
- 5) Quel est la différence essentiel entre une fonction et une procédure ?
- 6) Comment faire l'appel d'une fonction et celle d'une procédure

Exercice 2 :

Donner la syntaxe de création d'un tableau de 15 réels, et y ajouter les chiffres suivant dans l'ordre : 0, 1, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28;

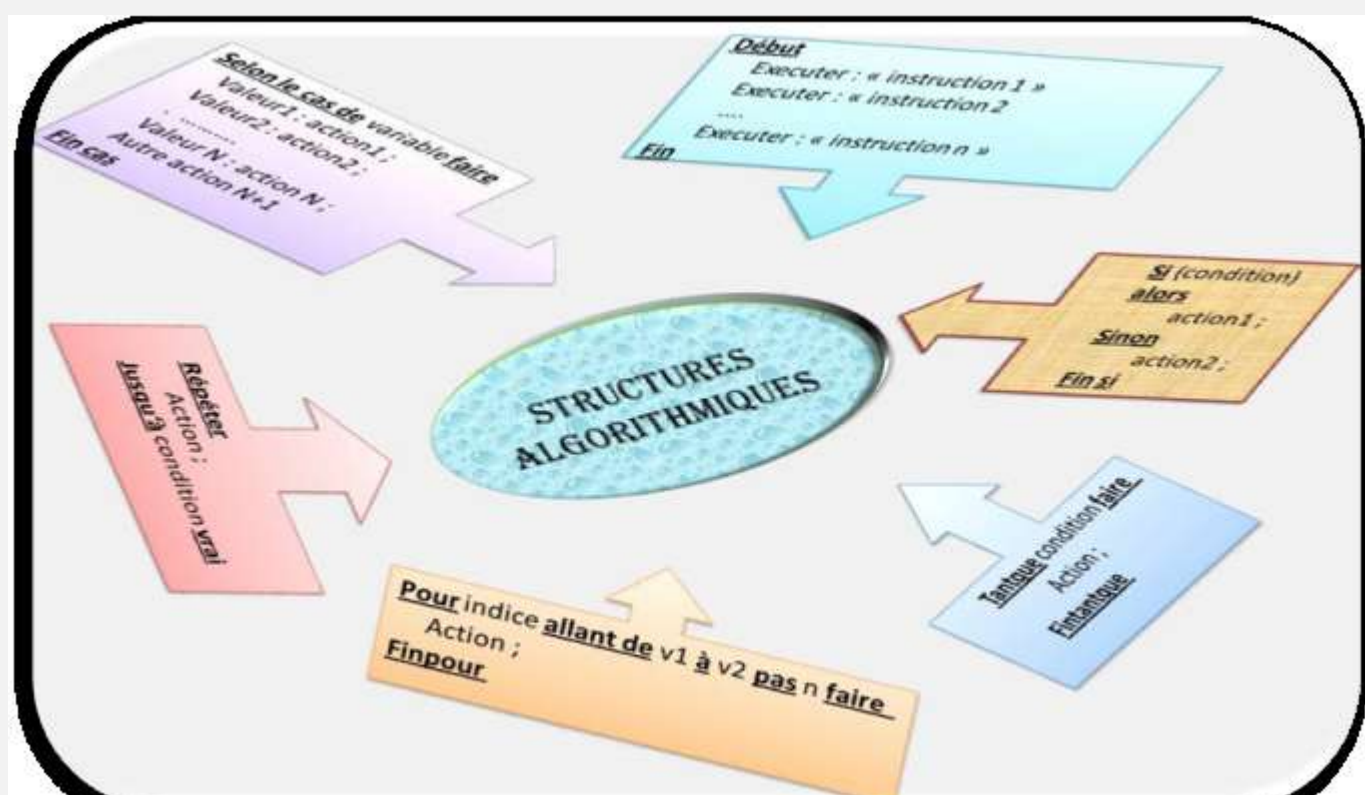
Exercice 3 :

Créer un tableau nommé Tab1 de taille 5 et remplit chaque case par son numéros élevés au cube. Créer un autre nommé Tab2 toujours de taille 5 et remplit chaque case par son numéro élevés au carré.

Modifier le contenu de Tab1 par le résultat de sa soustraction avec le tableau Tab2

Chapitre 7

LES STRUCTURES ALGORITHMIQUES



SOMMAIRE



- I- LES STRUCTURES SÉQUENTIELLES
- II- LES STRUCTURES ALTERNATIVES
- III- LES STRUCTURES DE CHOIX
- IV- LES STRUCTURES ITÉRATIVES OU RÉPÉTITIVES



Problème

Pendant la remise des copies d'informatique par votre enseignant, celui ci demande aux élèves d'écrire deux algorithmes afin d'améliorer chacun sa note.

Le premier algorithme est celui qui permet à votre enseignant, qu'en remettant les feuilles, il donne un stylo à tous ceux qui ont une note supérieure ou égale à 10/20 et punit tous ceux qui ont une note inférieure à 10/20.

Le deuxième algorithme est celui qui permet à votre enseignant de corriger les 90 copies de composition des élèves de la classe. En plaçant chaque instruction sur une ligne, cet algorithme doit tenir au plus sur 10 lignes.

- ⊕ Comment procéderiez-vous ?
- ⊕ Quelles structures algorithmiques allez-vous utiliser?



Pré-requis

- Connaître les étapes de résolution d'un problème
- Connaître la structure de base d'un algorithme
- Savoir manipuler une variable (déclaration, affectation, lecture et écriture à l'écran)



Objectifs

- Identifier les structures algorithmiques ;
- Donner la syntaxe de chaque structure
- Pouvoir intégrer les structures dans un algorithme
- Savoir choisir la structure appropriée pour la résolution d'un problème

INTRODUCTION

Les instructions dans un **algorithme** sont exécutées généralement de manière **séquentielle**. Par contre la puissance et le « comportement intelligent » d'un algorithme proviennent essentiellement :

- ⓐ De la possibilité de se comporter différemment suivant des circonstances, par exemple en fonction d'une réponse d'utilisateur, d'un résultat de calcul...
- ⓐ De la possibilité de répéter plusieurs fois un ensemble donné d'instructions.

Les opérations élémentaires relatives à la résolution d'un problème, peuvent en fonction de leur **enchaînement**, être organisées suivant une famille de **structures algorithmiques** fondamentales :

- ⓐ Les structures séquentielles ;
- ⓐ Les structures alternatives ;

- Ⓢ Les structures de choix ;
- Ⓢ Les structures itératives ou répétitives.

I. LES STRUCTURES SÉQUENTIELLES

Les **structures séquentielles** se caractérisent par une suite d'actions à exécuter successivement dans l'ordre énoncé. On utilisera le séparateur **point-virgule (;)** entre les actions à exécuter.

Une structure séquentielle peut donc se présenter de la manière suivante :

Début

Exécuter : « instruction 1 »

Exécuter : « instruction 2

....

Exécuter : « instruction n »

Fin

Exemple : Écrire un algorithme qui nous permet de laver l'OKOK.

Algorithme lavage de l'okok

début

Puise de l'eau ;

Mettre l'OKOK dans un seau ;

Verser de l'eau sur l'OKOK ;

Frotter l'OKOK avec des mains propres ;

Fin

II. LES STRUCTURES ALTERNATIVES

L'instruction **si** permet de programmer une structure dite de **choix** (ou **alternative**), permettant de **choisir entre deux instructions ou plus**, suivant la valeur d'une expression jouant le rôle de **condition**. La seconde partie, introduite par le mot clé **SINON**, est facultative, de sorte que l'instruction **Si** présente deux formes. On distingue **deux types de structures alternatives** :

1. La structure alternative complète

Dans cette structure **l'exécution** d'un des deux traitements distincts ne dépend que du **résultat d'un test** effectué sur la **condition** qui peut être une variable ou un événement.

Sa **syntaxe** est la suivante :

Si (condition) **alors**

action1 ;

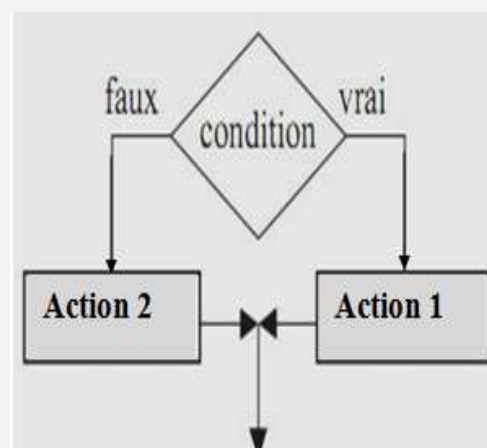
Sinon

action2 ;

Fin si

Ceci signifie que :

- Ⓢ Le test de la condition est effectué
- Ⓢ Si le test est positif, **action1** est exécuté
- Ⓢ Si le test est négatif, **action2** est exécuté



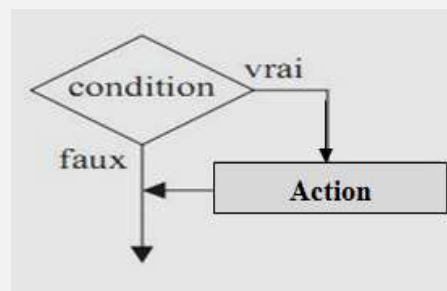
- ⓐ Sortir de la structure

2. La Structure alternative réduite

La structure alternative réduite se distingue de la précédente par le fait que seule la situation correspondant à la validation de la condition entraîne l'exécution du traitement, l'autre situation conduisant systématiquement à la sortie de la structure.

Sa **syntaxe** est la suivante :

Si (condition) **alors**
 Action ;
Fin si



Ceci signifie que :

- ⓐ Le test de la condition est effectué
- ⓑ Si le test est positif, **action** est exécutée
- ⓒ Sortir de la structure

Exemple 1:

Afficher ('nombre :') ;
 Saisir (nombre) ;
Si (nombre = 0) **alors**
 Afficher ('NULLE') ;
Sinon
 Afficher ('NON NULLE') ;
Fin si

Exemple 2:

Afficher ('nombre :') ;
 Saisir (nombre) ;
Si (nombre = 0) **alors**
 Afficher ('NULLE') ;
Fin si



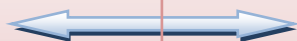
1. Condition est une expression dont la valeur de retour est un **booléen** (vrai ou faux)
2. Les **parenthèses** entourant l'expression jouant le rôle de **condition** font partie de la syntaxe de l'instruction et sont donc **obligatoires** ;
3. Les instructions figurant dans chaque partie du choix d'une structure **SI** sont absolument quelconques.
4. On peut également utiliser les **boucles SI imbriquées** les unes dans les autres. En d'autres termes, on peut avoir une **boucle SI** dans une autre **boucle SI** et ainsi de suite.

III. STRUCTURES DE CHOIX

La structure de choix permet, en fonction de **plusieurs conditions de type booléen**, d'effectuer des actions différentes suivant les valeurs que peut prendre une même variable. Elle se présente sous la forme suivante :

Selon le cas de variable faire

Valeur₁ : action₁ ;
 Valeur₂ : action₂ ;
 .
 .
 Valeur N : action N ;
 Autre action N+1
Fin cas



Si (variable = valeur₁) alors
 action₁ ;
sinon
Si (variable = valeur₂) alors
 Action₂ ;
sinon
si (variable = valeur₃) alors
 action₃ ;
sinon
 .
 .
Si (variable = valeur N) alors
 Action N ;
sinon
 action N+1 ;
Fin si
Fin si

Exemple :

Selon le cas de date faire

11 février : fêter la fête de la jeunesse ;
 20 mai : fêter la fête nationale ;
 25 décembre : fêter Noël ;
 14 février : fêter la sainte-valentin ;
 Autre : ne rien faire.

Fin cas

REMARQUE

1. Il faut veiller à ce que les valeurs des **sentinelles** de garde **valeur₁**, **valeur₂**, ..., **valeur_N** soient toutes différentes. Par contre les actions **action₁**, **action₂**, ..., **action_N**, **action_{N+1}** ne sont pas forcément différentes.
2. **Action 1**, **Action 2**, ..., **Action n** peuvent être des séquences d'instructions contenant à leurs tours d'autres structures de choix.

IV. LES STRUCTURES ITÉRATIVES

La structure itérative ou boucle répète l'exécution d'une instruction ou d'un ensemble d'instructions. Sa forme dépend de la connaissance qu'on a du nombre de répétition de l'action. On distingue deux cas.

1. Le nombre de répétitions n'est pas connu ou est variable

On distingue deux structures de bases :

- ⊗ La structure RÉPÉTER JUSQU'A
- ⊗ La structure TANT QUE FAIRE

a. La structure REPETER...JUSQU'A

Dans la structure **REPETER...JUSQU'A**, l'action est exécutée une première fois puis sa répétition se poursuit jusqu'à ce que la condition soit vérifiée. Elle se présente sous la forme suivante :

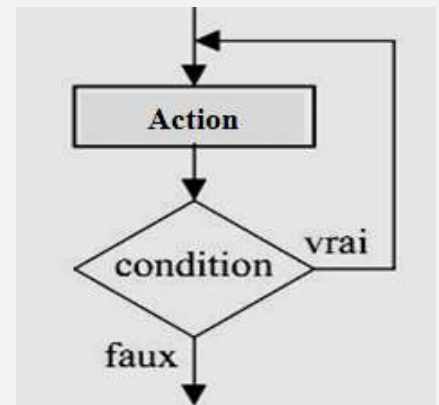
Répéter

Action ;

Jusqu'à condition vrai

La sémantique d'exécution de cette instruction est la suivante

- ⊗ Exécuter l'action
- ⊗ Évaluer la condition
- ⊗ Si la condition est vérifiée alors aller à la suite, sinon ré-exécuter l'action
- ⊗ Sortir de la structure



REMARQUE

L'action est exécutée au moins une fois.

b. La structure TANT QUE FAIRE

Dans la structure **TANT QUE...FAIRE**, on commence par tester la condition ; si elle est vérifiée, on exécute l'action. Cette structure se présente sous la forme suivante :

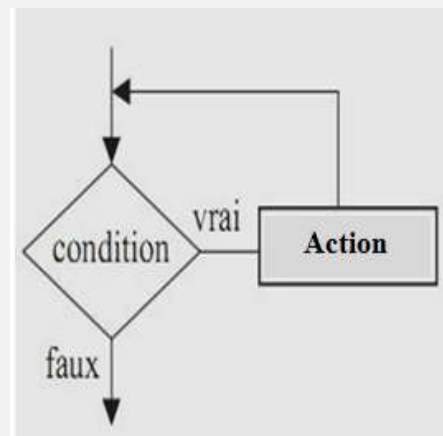
Tant que condition faire

Action ;

Fin tant que

La sémantique d'exécution de cette instruction est la suivante :

- ⊗ Évaluer la condition
- ⊗ Si la condition est vérifiée alors exécuter l'action
Sinon, aller à la ligne suivante (après la boucle)
- ⊗ Sortir de la structure





L'action peut ne jamais être exécutée



Répéter

Action ;

Jusqu'à (condition) vrai



Action ;

Tant que (condition) faire

Action

Fin tant que

Exemple 1:

Algorithme affichage

Variable nombre : Entier

debut

Repeter

Afficher (nombre) ;

Jusqu'à (nombre > 5) vrai

fin

Exemple 2:

Algorithme affichage

Variable nombre : Entier

debut

Afficher ('entrer nombre :) ;

Saisir (nombre) ;

Tantque (nombre ≤ 5) faire

Afficher (nombre) ;

Fin tant que

fin

c. Le nombre de répétition de l'action est connu

Dans cette structure, la sortie de la boucle d'itération s'effectue lorsque le nombre souhaité de répétition est atteint. On utilise donc une **variable** (ou indice) **de contrôle d'itération** (**i**) de la boucle caractérisée par sa **valeur initiale**, sa **valeur finale**, son **pas de variation**.

La structure de cette boucle est la suivante :

Pour indice allant de **v_1** à **v_2** pas **n** faire

Action ;

Fin pour

Deux cas de figure se présentent :

- ⊗ La valeur initiale de l'indice **v_1** est inférieure à la valeur finale de l'indice **v_2** : on parle de « **Pour croissant** ».
- ⊗ La valeur initiale de l'indice **v_1** est supérieure à la valeur finale de l'indice **v_2** : on parle de « **Pour décroissant** ».

REMARQUE

Lorsque rien n'est dit sur le pas, on considère qu'il vaut un. Le "pour croissant" est beaucoup plus utilisé.

ASTUCE

Pour i allant de 0 à 9 Pas de 1 faire
Afficher(i) ;
Fin pour



$i \leftarrow 0$;
Répéter
Afficher(i) ;
 $i \leftarrow i + 1$;
Jusqu'à (i = 9) vrai



Résumé

Les **instructions** d'un algorithme sont écrites généralement de façon **séquentielle**, mais il peut arriver qu'on souhaite représenter des situations faisant appel aux **choix**, aux **répétitions**... Pour palier à ce problème les opérations élémentaires, peuvent en fonction de leur enchainement, être organisées suivant une **famille de structures algorithmiques** que sont les **structures séquentielles**, les **structures alternatives**, les **structures de choix**, les **structures itératives ou répétitives**.

Les structures séquentielles sont caractérisées par l'écriture successive des actions à effectuer dans un ordre bien défini.

Les structures alternatives sont utilisées lorsqu'il existe plusieurs possibilités parmi lesquels une et une seule doit être choisie suivant une condition donnée.

Les structures de choix permettent, en fonction de plusieurs conditions de type booléen, d'effectuer des actions différentes suivant les valeurs que peut prendre une même variable.

Les structures itératives sont utilisées dans les situations où une même action est effectuée plusieurs fois.

ACTIVITÉS

ÉNONCÉ DE L'ACTIVITÉ

M. NGONGANG votre professeur de Mathématiques remet vos feuilles d'examen en suivant la position de chaque élève. Vous êtes au nombre de 150 dans votre salle de classe. On suppose que le travail du professeur est de **lire le nom de l'élève**, **relever la note de l'élève** et **remettre la copie**. Il doit effectuer ces opérations pour chaque élève jusqu'au dernier. En prenant pour opérations primitives : **lire_nom_élève**; **relever_note_élève**; **remettre_copie_élève**, écrire un algorithme nommé **remise_copie** qui permettra à l'enseignant de faire tout ce travail en utilisant

- ✓ La boucle pour ;
- ✓ La boucle tant que ;
- ✓ La boucle répéter.

SOLUTION :**1^{er} cas : la boucle pour**

On connaît le **nombre d'itération** qui est égal au nombre d'élève **150** ; Le professeur va commencer par le premier élève qui porte le numéro **1** donc la **valeur initiale est 1**. L'enseignant doit distribuer les copies à tous les élèves jusqu'au dernier qui porte le numéro **150**. **La valeur finale est de 150**. Puisqu'on va d'un élève à un autre en suivant des numéros consécutifs ($N^o 1, 2, \dots, 150$) **le pas est de 1**

Algorithme remise_copie

Variable numéro : Entier ;

Debut

Pour numero **allant de 1 à 150 pas de 1 faire**

Lire nom élève ;

Relever note élève ;

Remettre copie élève ;

Fin pour

fin

2^{ème} cas : boucle TANT QUE

Avec cette boucle, la **condition est testée** au préalable **avant l'exécution des instructions** qui se trouvent à l'intérieur de la boucle. Pour qu'on puisse entrer dans l'algorithme au moins une fois pour effectuer le traitement, il faut **initialiser la variable** de sorte qu'elle respecte la condition d'entrée et exécuter une première fois les instructions.

3^{ème} cas : boucle REPETER

Dans ce cas par contre, la condition est testée à la fin, ce qui fait en sorte que les instructions à l'intérieur de la boucle sont exécutées au moins une fois. On n'a donc pas besoin d'exécuter les instructions avant d'entrer dans la boucle.

Algorithme remise_copie

Variable numéro : Entier ;

Debut

Numero $\leftarrow 1$

Lire nom eleve ;

Relever note eleve ;

Remettre copie eleve ;

Tant que (numero ≤ 150) **faire**

Numero $\leftarrow$ Numero + 1 ;

Lire nom eleve ;

Relever note eleve ;

Remettre copie eleve ;

Fin tant que

fin

Algorithme remise_copie

Variable numéro : Entier ;

Debut

Numero $\leftarrow 0$;

Repeter

Lire nom eleve ;

Relever note eleve ;

Remettre copie eleve ;

Numero $\leftarrow$ Numero + 1 ;

Jusqu'à (numero > 150) **vrai**

fin



EXERCICE 1:

- 1) Quelles sont les quatre familles de structures algorithmiques ?
- 2) Quelles sont les caractéristiques d'une structure séquentielle ?
- 3) Donner les deux types de structures alternatives et leur syntaxe.

EXERCICE 2:

- 1) Quelle différence y a-t-il entre les boucles REPETER et TANT QUE ?
- 2) Quels sont les données indispensables à l'utilisation de la boucle POUR ?
- 3) Donner les structures des boucles POUR, TANT QUE et REPETER.

EXERCICE 3:

Votre papa décide de vous primer par rapport à la fin d'année selon votre mention. Pour le faire il vous impose les conditions suivantes :

- ✓ Pour la mention p (passable) il vous achète des bonbons ;
- ✓ Pour la mention AB il vous offre un vélo ;
- ✓ Pour la mention B Il vous offre une console de jeux ;
- ✓ Pour la mention TB il vous offre un billet d'avion pour Paris.

Écrire un algorithme qui permet d'effectuer cette opération en utilisant :

- ✓ Les structures de choix ;
- ✓ Les structures conditionnelles.

On utilisera les opérations primitives suivantes : acheter_bonbon, offrir_vélo, offrir_console, offrir_billet pour les opérations respectives décrites plus haut.

EXERCICE 4:

Écrire l'algorithme permettant d'afficher à l'écran tous les nombres impairs inférieurs à 200. Utiliser la boucle POUR et TANT QUE.

EXERCICE 5:

Écrire l'algorithme qui prend en entrée deux nombres et affiche le plus grand.

EXERCICE 6:

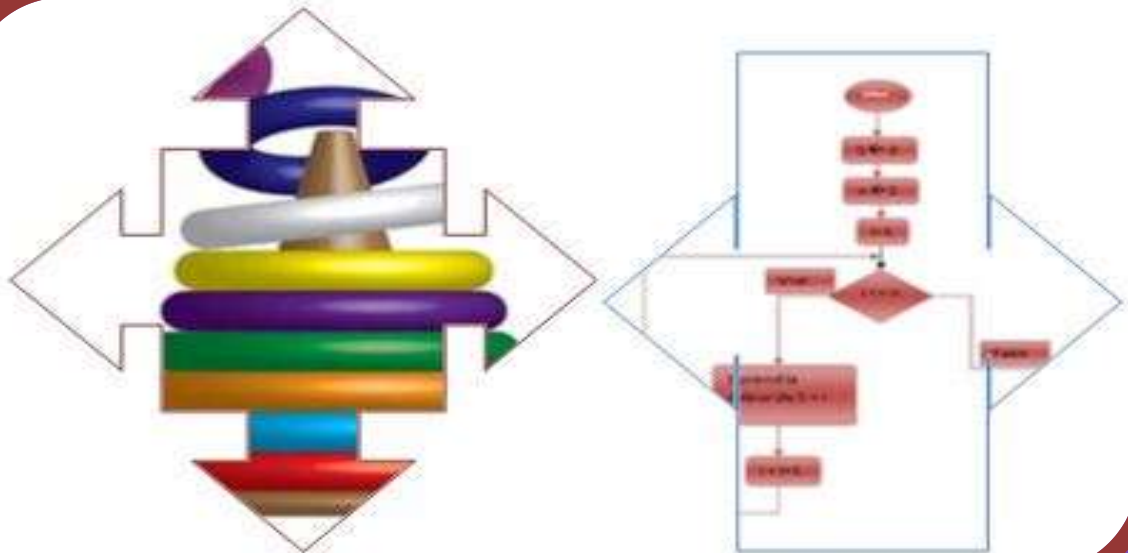
Que produit l'algorithme suivant :

```

algorithme
    Variable N :Entier
    Debut
        N ← 0 ;
        Repeter
            N ← N + 3 ;
        Jusqu'à (N > 45) vrai
    fin
  
```

Chapitre 8

EXÉCUTION D'UN ALGORITHME



SOMMAIRE



- I- ÉTAT DES VARIABLES AU COURS DE L'EXÉCUTION ET EXÉCUTION DE QUELQUES ALGORITHMES
- II- DÉCOMPTÉ DU NOMBRE D'INSTRUCTION ET CONSTRUCTION D'UN ORGANIGRAMME



Problème

Les algorithmes sont une façon de formuler la résolution d'un problème. Votre mère est commissaire au compte dans sa réunion ; elle aimerait connaître l'état d'avancement de l'épargne de chaque membre de la tontine. Votre connaissance en algorithmique vous a permis d'écrire l'algorithme qui calcul les intérêts cumulés en fonction du taux d'intérêt et des placements.

Comment allez-vous procéder pour satisfaire les attentes de votre mère ? C'est-à-dire lui présenter les intérêts par mois et par participant.



Pré-requis

- Connaître le rôle d'un algorithme
- Connaître la structure d'un algorithme
- Être capable d'écrire un algorithme



Objectifs

- Exécuter à la main un algorithme
- Décompter les instructions d'un algorithme
- Construire l'organigramme d'un algorithme

INTRODUCTION

L'algorithme est une suite de règles opératoires à appliquer dans un ordre déterminé à un nombre fini de données afin d'effectuer un calcul numérique en un nombre fini d'étapes. Ces règles opératoires sont définies en fonction du problème que l'on veut résoudre et de façon naturelle, l'enchaînement de ces règles ou instructions déterminent la solution au problème à résoudre.

Dans votre quotidien, il n'existe pas un instant que vous passez sans exécuter un algorithme. Vous exécutez un algorithme quand vous vous apprêtez pour l'école, quand vous faites la vaisselle ou la lessive, quand vous faites la cuisine, etc.... Mais vous vous demandez comment ? Ce chapitre consiste à vous faire connaître et comprendre comment exécuter un algorithme.

I- ÉTAT DES VARIABLES AU COURS DE L'EXÉCUTION DE QUELQUES ALGORITHMES

1. État des variables au cours de l'exécution d'un algorithme

Les variables sont des emplacements mémoire qui contiennent des valeurs précises à une étape précise de l'exécution d'un algorithme. Elles permettent de stocker les résultats

temporaires ou définitifs des calculs effectués. L'état d'une variable est en quelque sorte la valeur de la variable à un moment donné.

Au cours de l'exécution d'un algorithme, l'état des variables varie en fonction des instructions qui prennent en compte ces variables.

2. Exécution d'algorithme

a. Instruction d'entrée-sortie et d'affectation



ÉNONCÉ : Considérons l'algorithme suivant décrit en langage naturel :

- Donner à x une valeur
- Calculer $u = x + 4$
- Calculer $y = u \times x$
 - Que renvoie l'algorithme pour $x=5$? $x=10$?
 - L'algorithme définit une fonction f . Exprimer $f(x)$ en fonction de x

Il faut noter que $+$ est le signe de l'addition et $*$ le signe de la multiplication. La fonction `afficher ()` est primitive et permet d'afficher à l'écran le contenu de la variable prise en paramètre.

L'écriture codée de cet algorithme est la suivante :

Algorithme EntréeSortie_Affectation

Variable Entier : x, u, y ;

Procédure

`afficher (Type t) ; /* affiche à l'écran l'élément t de type Type */`

Début

u prend la valeur de $x + 4$;

y prend la valeur de $u \times x$;

`afficher (y) ;`

FinAlgo

Solution

i. Si $x = 5$, l'algorithme calcule $u = x + 4 = 5 + 4 = 9$ puis affiche la valeur $y = u \times x = 9 \times 5 = 45$

Si $x = 10$, l'algorithme calcule $u = x + 4 = 10 + 4 = 14$ puis affiche la valeur $y = u \times x = 14 \times 10 = 140$

ii. Pour toute valeur de x , l'algorithme calcule $u = x + 4$ puis affiche la valeur $y = u \times x = (x + 4) \times x = x^2 + 4x$

L'algorithme définit donc une fonction f telle que pour tout x , $f(x) = x^2 + 4x$

b. Structure alternative

**ÉNONCÉ :**

Un polygame décide après plusieurs disputes dans son foyer de donner de l'argent à ses multiples femmes d'après le principe suivant :

- Si l'ordre d'arrivée de la femme dans le foyer est un nombre entier pair alors elle reçoit par jour $\frac{\text{ordre}}{2} * 1000 \text{ Francs cfa}$
- Si l'ordre d'arrivée de la femme dans le foyer est un nombre entier impair alors elle reçoit par jour $(3 * \text{ordre} + 1) * 1000 \text{ Francs cfa}$

Ce problème peut être formulé algorithmiquement en langage naturel comme suit :

- Donner à *ordre* une valeur
- Tester la parité de *ordre*
- Calculer $t = \frac{\text{ordre}}{2}$
- Si *t* est entier *rst* prend la valeur 0
- Si *t* n'est pas entier *rst* prend la valeur 1
- Si *rst* vaut 0 alors calculer $y = \frac{\text{ordre}}{2} * 1000$
- Si *rst* vaut 1 calculer $y = (3 * \text{ordre} + 1) * 1000$

Evaluer grâce à l'algorithme ce que le polygame donne à ses femmes d'ordres respectifs 1, 2, 3, 4, 50, 51.

Il faut noter que + est le signe de **l'addition**, --- le signe de la **division** et * le signe de la **multiplication**

L'écriture codée de cet algorithme est la suivante :

Algorithme alternation

Variable Entier : ordre, rst ;

Réel : t, y ;

Fonction

Entier estEntier (Type e) ;

/* renvoie 0 si e est de type Entier et 1 sinon */

Procédure

afficher (Type t) ;

//affiche à l'écran l'élément t de type Type

Début

t prend la valeur de $\frac{\text{ordre}}{2}$;

Si estEntier(t) == 0 alors

rst prend la valeur de 0 ;

Sinon

rst prend la valeur de 1 ;

FinSi

Si rst == 0 alors

// ordre est un entier pair

y prend la valeur de $(\frac{\text{ordre}}{2}) * 1000$;

1000 ;

Sinon

Si rst == 1 alors

y prend la valeur de $(3 * \text{ordre} + 1) * 1000$;

FinSi

FinSi

afficher (y) ;

Fin

Solution

Pour ordre=1 $t = \text{ordre}/2 = 1/2 = 0,5$
 $\text{estEntier}(0,5) \rightarrow 1$ donc $\text{rst} = 1$
 $y = (3 * \text{ordre} + 1) * 1000 = (3 * 1 + 1) * 1000 = 4000$

Affichage 4000

Pour ordre=2 $t = \text{ordre}/2 = 2/2 = 1$
 $\text{estEntier}(1) \rightarrow 0$ donc $\text{rst} = 0$
 $y = (\text{ordre}/2) * 1000 = (2/2) * 1000 = 1000$

Affichage 1000

APPLICATION : Exécuter l'algorithme pour la suite des valeurs de *ordre*

c. Structure itérative

ÉNONCÉ : Considérons l'algorithme suivant décrit en langage naturel :

Choisir un entier naturel **non nul** n

- **Initialiser** une variable S en lui affectant la valeur 0
 - **Répéter** pour chaque entier i de 1 jusqu'à n l'instruction : « Donner à S la valeur $S+i$ »
 - à la fin de la répétition, **afficher** S .
- i- Que renvoie l'algorithme pour $n=10$?
 - ii- Quel est le rôle de cet algorithme ?

L'écriture codée de cet algorithme est la suivante :

Algorithme itération

Variable Entier : n, i, S ;

Procédure

afficher (Type t) ;
 /* affiche à l'écran l'élément t de type */
 Type
 lire (Type t) ;
 /* lit une valeur de type Type et met dans
 la variable t */

Début

S prend la valeur 0 ;
 lire (n) ;
Pour i de 1 jusqu'à n **faire**
 S prend la valeur de $S+i$;
finPour
 afficher (S) ;
FinAlgo

Solution

i- **Si** $n=10$

$S=0$
 $i=1, S=S+i=0+1=1$
 $i=2, S=S+i=1+2=3$
 $i=3, S=S+i=3+3=6$
 $i=4, S=S+i=6+4=10$
 $i=5, S=S+i=10+5=15$

$i=6, S=S+i=15+6=21$
 $i=7, S=S+i=21+7=28$
 $i=8, S=S+i=28+8=36$
 $i=9, S=S+i=36+9=45$
 $i=10, S=S+i=45+10=55$
 Fin de l'algorithme, on affiche $S=55$

On peut aussi présenter cet algorithme sous la forme d'un tableau :

	Valeur de i	Valeur de S
Initialisation	0	0
Etape 1	1	$0+1=1$
Etape 2	2	$1+2=3$
Etape 3	3	$3+3=6$
Etape 4	4	$6+4=10$
Etape 5	5	$10+5=15$
Etape 6	6	$15+6=21$
Etape 7	7	$21+7=28$
Etape 8	8	$28+8=36$
Etape 9	9	$36+9=45$
Etape 10	10	$45+10=55$
Affichage		55

- ii- Cet algorithme calcule la **somme des n premiers entiers consécutifs de 1 à n** , en stockant successivement la somme dans la variable S .



Toute l'exécution d'un algorithme se joue au niveau de ses variables (variables de boucle ou du programme). Il faut de ce fait toujours vérifier que les variables sont bien positionnées à chaque étape.

II. DÉCOMPTÉ DU NOMBRE D'INSTRUCTIONS ET CONSTRUCTION D'UN ORGANIGRAMME

I. Décompte des instructions ou Complexité d'un algorithme

Compter le nombre d'instruction d'un algorithme revient à calculer la complexité en temps de cet algorithme. Ce nombre n'est pas toujours exact ; en effet le calcul de la complexité dépend du type d'opération que l'on considère. Les opérations choisies sont appelées les opérations fondamentales.

La première étape dans le décompte des instructions est la détermination des **opérations fondamentales**.

La deuxième étape est le **décompte** proprement dit. Il se fait de façon simple pour les algorithmes non récursifs et n'utilisant pas de structures itératives. Dans ce cas de figure, il suffit juste de compter le nombre d'instructions fondamentales.

Dans le cas des algorithmes ayant des structures itératives, il faudra compter le nombre d'opérations fondamentales pour un tour de boucle et multiplier par le nombre de fois que l'on itère.

Dans le cas des algorithmes récursifs, il faut calculer le nombre d'opération par appel de fonction et en fonction des paramètres d'appel ; puis estimer le nombre d'appel et faire le calcul pour chaque appel récursif ; enfin additionner tous les résultats intermédiaires pour obtenir la complexité. Dans beaucoup de cas, on aboutit à des fonctions mathématiques qu'il faut résoudre.

Illustration

Reprenons l'exemple de notre algorithme nommé *itération*.

Choisissons comme opérations élémentaires l'addition et l'affectation. Dans l'algorithme nous avons une structure itérative qui est le *Pour*.

A l'intérieur de cette structure, nous avons deux instructions à savoir : une instruction d'affectation et une instruction d'addition. Il faut maintenant calculer le nombre de fois que l'on passe dans la boucle *Pour*. Au vue des arguments, on y passe n fois. On aura donc $2 \times n$ opérations réalisées à la fin de la boucle.

A l'extérieure de cette structure, nous avons une instruction d'affectation (*S prend la valeur o*).

Au total nous avons $2 \times n + 1$ opérations dans notre algorithme

2. Construction d'un organigramme

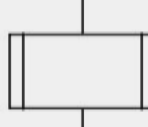
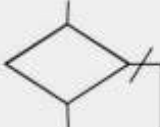
Beaucoup de conventions ont été proposées pour visualiser les algorithmes. Elles se situent le plus souvent au niveau des constructions de base. Quelques unes de ces méthodes sont :

- ⊙ L'organigramme
- ⊙ Le diagramme de Sassi-Shneidermann
- ⊙ L'arbre programmatique
- ⊙ Le pseudo-code

Nous allons étudier la première convention à savoir l'organigramme.

L'organigramme est une représentation graphique de la structure d'une entreprise, d'un algorithme ou d'un module. Il permet de visualiser l'enchaînement des calculs effectués par la machine. L'organigramme permet également de mieux appréhender l'évolution de l'exécution d'un algorithme.

a. Quelques symboles pour la construction d'un organigramme

Symbole	Signification	Symbole	Signification
	Symbole général : instruction ou groupe d'instruction n'ayant pas de symbole normalisé		Liaison : symbole pour assurer la continuité lorsqu'une partie de ligne de liaison n'est pas représentée
	Sous-programme : utilisé pour les appels de fonctions		Début, fin, interruption d'un algorithme
	Affectation : mise à disposition d'une information à traiter ou enregistrement d'une information traitée.		Commentaire : utiliser pour donner des indications sur les opérations effectuées.
	Branchement : exploitation de conditions variables impliquant un choix parmi plusieurs.		

b. Organigramme de la structure Pour ...Faire

Cette structure permet d'exécuter une séquence pour un certain nombre de valeur.

La syntaxe s'emploie sous la forme :

Pour (arg1 ; arg 2 ; arg 3) **Faire**
Instruction 1 ;

...

Instruction n ;

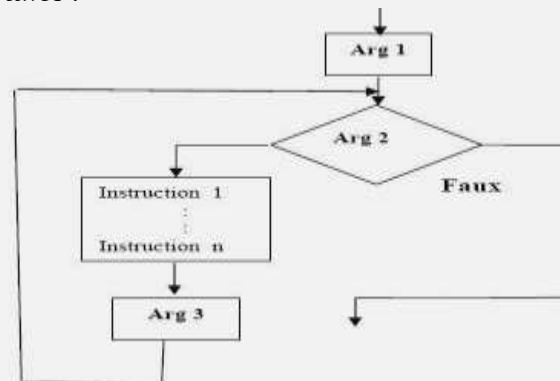
FinPour

arg1 représente l'initialisation du compteur

arg 2 représente la condition d'arrêt

arg 3 représente l'incrément ou la décrémentation du compteur

L'organigramme associé à cette syntaxe est la suivante :



ÉNONCÉ : Concevoir l'organigramme de l'algorithme ci- dessous

Algorithme itération

Variable Entier : n, i, S ;

Procédure

afficher (Type t) ;

//affiche à l'écran l'élément t de type

Type

lire (Type t) ;

//lit une valeur de type Type et met dans la variable t

Début

S prend la valeur 0 ;

lire (n) ;

Pour i de 1 jusqu'à n **faire** // pour (i=1, i ≤ n, i=i+1)

S prend la valeur de S+i ;

finPour

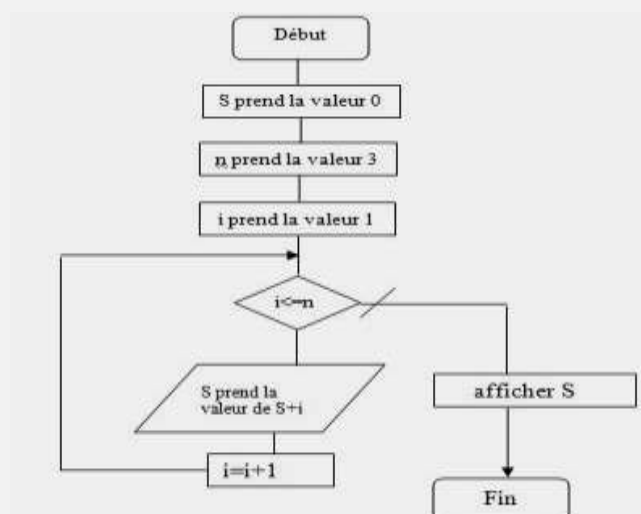
afficher (S) ;

FinAlgo

arg1 -> i=1 ; arg2 -> i ≤ n ; arg3 -> i=i+1

- ⊙ Tout d'abord, s'inspirer de l'organigramme de base de la boucle **pour**
- ⊙ Mentionner le début de l'algorithme par un cercle
- ⊙ Mettre les instructions précédant l'itération dans de petits rectangles
- ⊙ Ajouter le diagramme pour la boucle et marquer la fin par un cercle

A l'issu de ce cheminement, vous obtenez l'organigramme ci-après :





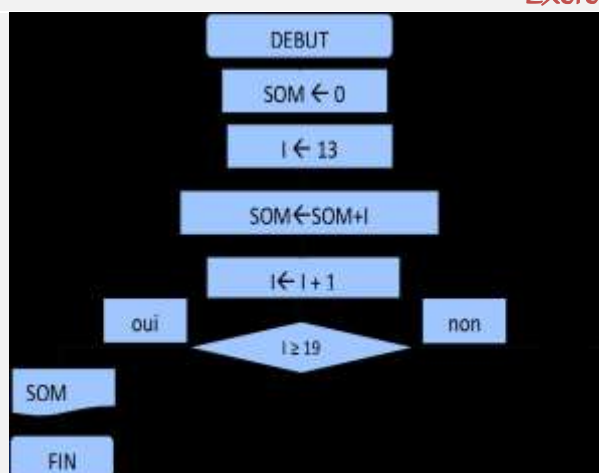
Résumé

Pour exécuter un algorithme il faut :

- Tout d'abord initialiser les variables. Chaque instruction modifie en général au moins une variable
- Identifier toutes les structures et se rassurer que les variables à modifier le sont vraiment à chaque étape



Exercice 1



Soit l'organigramme ci-contre :

1. Donner les différentes structures utilisées dans cet algorithme
2. Exécuter l'algorithme et observer la valeur de la variable SOM à chaque itération
3. Que fait cet algorithme ?

Exercice 2

En exécutant les instructions suivantes :

Var

entier A, B, C;

A ← 3; (1)

B ← 12; (2)

A ← B; (3)

B ← A; (4)

C ← A + B; (5)

Afficher(C); (6)

❖ La valeur affichée en sortie sera

a) 12 ☐ b) 15 ☐ c) 24 ☐ d) aucune réponse

n'est exacte. ☐

❖ Aux étapes (3) et (4), qu'advient-il des valeurs de A et B ?

❖ si l'on inverse les instructions (3) et (4), qu'advient-il des valeurs de A et B ?

Exercice 3

Votre mère madame KENGMO aimerait que vous l'aidiez à résoudre le problème ci-contre : un membre de la réunion a fait un placement x au mois de janvier et un placement y au mois de mars. Pendant un mois l'intérêt est au taux de 3% et les intérêts sont cumulés c'est-à-dire, un intérêt au mois i devient un capital au mois i+1. Donner l'évolution des placements du membre de la réunion pendant tous les mois jusqu'au mois de septembre où la caisse sera cassée.

- ❖ Écrire un algorithme pour chacune des structures Pour, Tant que, Répéter.
- ❖ Exécuter l'algorithme et afficher les résultats pour chaque mois.

Indication : chaque tour de boucle définira un mois. Pour chaque tour de boucle faites le calcul de l'intérêt de ce mois et du nouveau capital ; afficher ce capital et l'intérêt.

MODULE

3

CRÉATION D'UN SITE WEB

- 1- Notion de page web*
- 2- Insertion des liens, image et sons dans une page web*
- 3- Insertion des tableaux et des frames dans une page web*
- 4- Insertion des formulaires et des boutons dans une page web*
- 5- Production d'une feuille de style*

Chapitre 9

NOTION DE PAGE WEB



SOMMAIRE



- I. PRÉPARATION D'UN ENVIRONNEMENT DE PROGRAMMATION WEB
- II. NOTIONS DE BASE DU LANGAGE HTML
- III. LES BALISES



Problème

Il vous est demandé de réaliser le journal de votre famille. Ce journal consiste à donner une description de tous les membres de votre famille.

- ⊕ Dites comment vous allez procéder pour mettre sur pied votre journal.
- ⊕ Quels sont les éléments qu'on pourra retrouver sur une page de votre journal familial.



Pré-requis

- Connaître les étapes de résolution d'un problème
- Savoir segmenter un problème en différentes tâches
- Connaître la structure de base d'un algorithme



Objectifs

- Connaître les notions de base d'un site web
- Connaître l'utilité et le fonctionnement d'un site web
- Connaître la structure d'une page web
- Maîtriser les fonctions des balises généralement utilisées

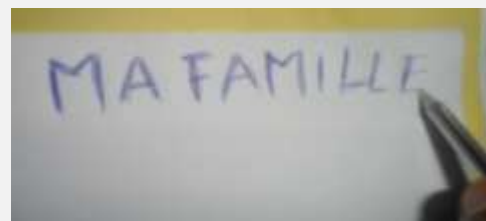
INTRODUCTION

Pour la réalisation d'un journal de famille, on a ensemble d'étapes par lesquelles on doit passer pour y arriver. Ces étapes peuvent être groupées en deux points qui sont les suivants :

- ⊙ Présenter brièvement l'histoire de votre famille suivie de l'énumération des différents membres qui la constitue. Ce travail se fera sur une feuille.
- ⊙ Présenter de manière détaillée chaque membre de la famille. Chaque membre de la famille sera présenté sur une feuille.

Étape 1 :

Prendre une feuille sur laquelle vous allez écrire à l'entête le titre de votre journal, soit « MA FAMILLE ». Écrire ce titre en gros caractères et en majuscule sur deux centimètres de hauteur. Cette feuille sera la page principal ou d'accueil de votre journal.



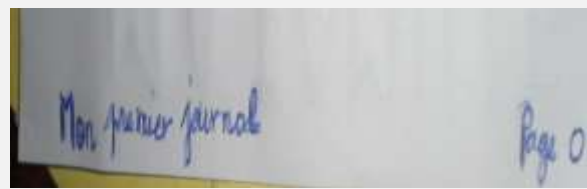
Étape 2 :

Sauter un espace d'environ trois à quatre centimètres (3 à 4 cm). Présenter brièvement l'histoire de votre famille. Allez à la ligne et énumérer de manière globale les différents membres de votre famille. Énumérer chacun sur une ligne seule. Ceci doit être contenu dans la même page. Écrire en caractère normal (moyen), en majuscule et/ou minuscule selon qu'il s'agisse d'un nom propre ou non, ou si c'est le début d'une phrase.



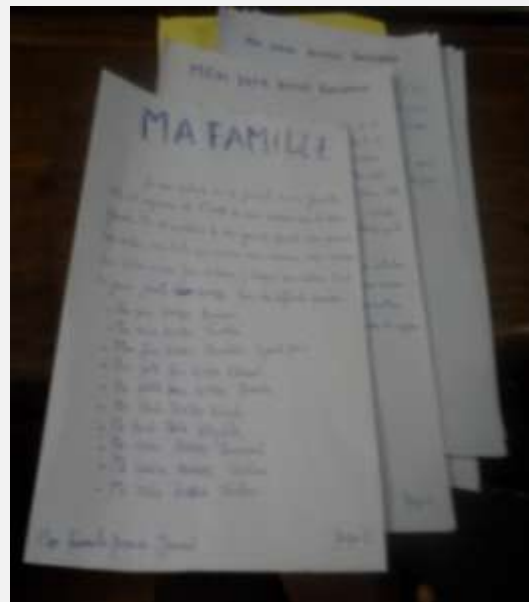
Étape 3 :

Allez à la fin de votre feuille du côté gauche et y écrire la phrase suivante : « **mon premier journal** ». Écrire cette phrase en petit caractère et en minuscule. Écrire « **page 0** » du côté droit.

**Étape 4 :**

Prendre une feuille autre, lui donner pour titre le nom du premier membre de famille cité à la « **page 0** ». Ce titre doit être écrit en gros caractères et en majuscule mais pas plus grand que le titre de la page d'accueil. Sauter un espace de trois centimètres environ. Décrivez la vie du membre de famille donc **le nom est porté sur le titre**.

Allez au bas de la feuille et faire comme à l'étape 3 en indiquant toujours à droite le numéro de la page correspondante qui dans ce cas sera « **page 1** ».

**Étape 5 :**

Reprendre l'étape 4 pour chaque membre de famille.

Étape 6 :

Revenir sur la page d'accueil et écrire à la suite du nom de chaque membre de famille le numéro de la page dans laquelle il est présenté. Par exemple, « **page 4** » à la suite du nom du membre de famille décrit à la quatrième page après la page d'accueil.

Ici image

Qu'est-ce que vous remarquez ?

En se basant sur cette activité, on remarque qu'un journal est un ensemble de feuilles mises ensemble. La **page d'accueil** donne une présentation globale du journal et indique les **numéros des pages** dans lesquelles sont développés des éléments se trouvant sur la page d'accueil. Ces numéros constituent des **liens** permettant d'identifier une page pour facilement y accéder. Chaque page possède un **entête** contenant un **titre**, un **corps** dans lequel est développé le sujet, et un **pied de page** qui contient une note de bas de page et le numéro de la page.

Cette organisation est similaire à celle d'un **site web**, qui est défini comme étant *est un ensemble de pages web liés par des liens permettant d'aller d'une page à l'autre*. La structure des pages web sont similaires à celles du journal. On retrouve les éléments suivants :

- ② Un **en-tête** de page qui contient le titre de la page web
- ② Un **corps** qui contient le contenu détaillé
- ② Un **pied de page** (éventuellement) qui contient des éléments supplémentaires.

On peut retrouver des liens permettant d'aller d'une page à l'autre dans toutes les parties d'une page web.

La création d'une page web est fonction du langage utilisé. Comme langage de création de page web, on a le **HTML, XHTML, CSS, JavaScript, Php...** Il est à noter que la structure d'une page web ne change pas quelque soit le langage utilisé.

Dans le cadre de ce cours, nous allons nous appuyer sur le langage **HTML**, qui est un langage universel utilisé pour communiquer sur le Web. C'est un langage de **description** (et non pas de programmation) qui permet de décrire l'aspect d'une page, d'y inclure des informations variées (textes, images, sons, animations etc.) et d'établir des relations cohérentes entre ces informations grâce aux liens hypertextes.

Un **site web** est un ensemble de pages web liés par des liens permettant d'aller d'une page à l'autre.

Une **page web** est la description d'un code Html. Le langage Html est donc à la base de la création des sites web.

Les **avantages** du langage HTML sont nombreux :

- ⊕ **peu coûteux** en effet, un simple éditeur de texte suffit à écrire ses premiers codes HTML ;
- ⊕ relativement **facile** à aborder et accessible même aux non initiés ;
- ⊕ il représente en outre un **bon moyen de dépasser les problèmes de compatibilité** entre des systèmes et des formats informatiques différents. Il s'adapte à toutes les plateformes que ce soit Windows, Macintosh, Unix, OS...

I- PRÉPARER L'ENVIRONNEMENT DE PROGRAMMATION WEB

1. Télécharger et installer un éditeur de texte

Un **éditeur de texte** est un programme dédié à l'écriture de code. On peut en général l'utiliser pour de multiples langages au delà du Html.

Un éditeur de texte simple comme par exemple le Bloc-notes ou Notepad de Windows suffit pour créer nos pages web. Mais, il existe des éditeurs de texte encore plus performants mais coûteux comme Claris Home Page, Frontpage de Microsoft, Hotdog, Hotmetal, WebExpert ou Netscape Editor... Il suffit donc de télécharger l'un d'eux et de l'installer dans votre ordinateur.

Le **téléchargement** et **l'installation** d'un logiciel a déjà été vu dans les classes antérieures. Il est juste question ici de donner quelques indications supplémentaires pour facilement télécharger et installer un logiciel.

- Ⓢ Plusieurs pages web proposent l'installation d'un même logiciel. Pour les recenser, il suffit de saisir le nom du logiciel dans la zone de recherche d'un moteur de recherche. Lire les indications données par les résultats de la recherche pour chaque page proposé afin de faire le meilleur choix de la page de téléchargement.
- Ⓢ A la fin du téléchargement de votre logiciel, lancer l'installation en double-cliquant sur l'icône apparaissant à la fin du téléchargement dans la zone où vous l'aviez enregistré. Il ne reste plus qu'à suivre les étapes classiques d'installation d'un logiciel en lisant évidemment les instructions avant de faire un quelconque choix.



Il faut noter que les logiciels propriétaires sont en général proposés en version d'essai n'ayant pas besoin d'une licence mais dont l'accès est limité dans le temps et en fonctionnalités ; et en version complète exigeant l'achat d'une licence. Faire le choix du logiciel au téléchargement ou à son installation selon les moyens dont vous disposez. Mais il existe aussi des logiciels libres et gratuits que vous pouvez télécharger sur Internet.

2. Utiliser un éditeur et choisir un navigateur pour notre page web

Un **navigateur** est un programme qui permet de surfer sur le Net et d'afficher sur un écran les pages web. Il y a beaucoup de marques et de types de navigateurs différents. Des simples, des archaïques ou des sophistiqués... Les plus connus sont Netscape, Internet Explorer, Opéra, Mozilla Firefox etc.

- ⊕ Chaque navigateur a sa propre façon de travailler et d'afficher une page web selon son propre mode d'affichage. Donc, le navigateur adapte votre page web par rapport à ses propres caractéristiques.
- ⊕ En conséquence, vous n'avez pas la maîtrise totale de votre document en Html.
- ⊕ Mais Html est un langage universel qui s'adapte à toutes les plateformes que ce soit Windows, Macintosh, Unix, OS...



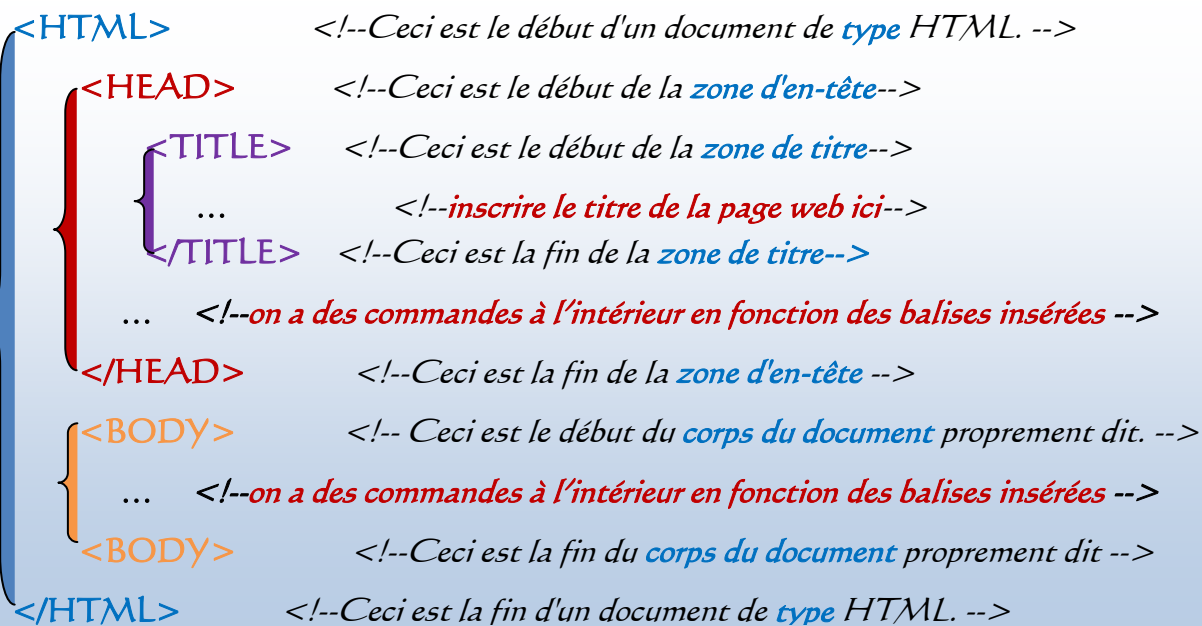
- ⊕ Pas besoin d'être connecté pour écrire, voir et peaufiner vos pages Html.
- ⊕ Si les éditeurs Html vous facilitent grandement la tâche, ils ne sont pas toujours parfaits surtout lors des modifications, annulations ou suppressions en cours de travail. Il vous faudra bien alors vous plonger dans le code source pour corriger les dysfonctionnements.
- ⊕ Les codes sources de vos pages préférées sont disponibles (et sans copyright). Il est alors possible de s'en inspirer pour reprendre le procédé sans avoir à réinventer le monde.

II- NOTIONS DE BASE DU LANGAGE HTML

1. Structure de base d'un document HTML

Le langage Html est basé sur des instructions comme tout langage algorithmique. En Html, les instructions sont introduites par des mots clés entourés par les signes **<** et **>** (**<mot clé>**), comme par exemple **<html>**. Ces mots clés sont appelés des **tags** ou des **balises**. Et ce sont ces balises qui servent à structurer un document Html.

Un document Html est de manière générale structuré selon le squelette suivant :



```

<HTML>      <!--Ceci est le début d'un document de type HTML. -->
{
  <HEAD>     <!--Ceci est le début de la zone d'en-tête-->
  {
    <TITLE>   <!--Ceci est le début de la zone de titre-->
    ...      <!--inscrire le titre de la page web ici-->
    </TITLE>  <!--Ceci est la fin de la zone de titre-->
  }
  ...       <!--on a des commandes à l'intérieur en fonction des balises insérées -->
  </HEAD>    <!--Ceci est la fin de la zone d'en-tête -->
  {
    <BODY>    <!-- Ceci est le début du corps du document proprement dit. -->
    ...       <!--on a des commandes à l'intérieur en fonction des balises insérées -->
    <BODY>    <!--Ceci est la fin du corps du document proprement dit -->
  }
</HTML>     <!--Ceci est la fin d'un document de type HTML. -->

```

2. Produire et tester un document HTML minimum

Faisons ensemble notre premier document Html en suivant les étapes suivantes :

- ⊕ Ouvrir l'éditeur de texte.
- ⊕ Écrire les codes Html suivants

```

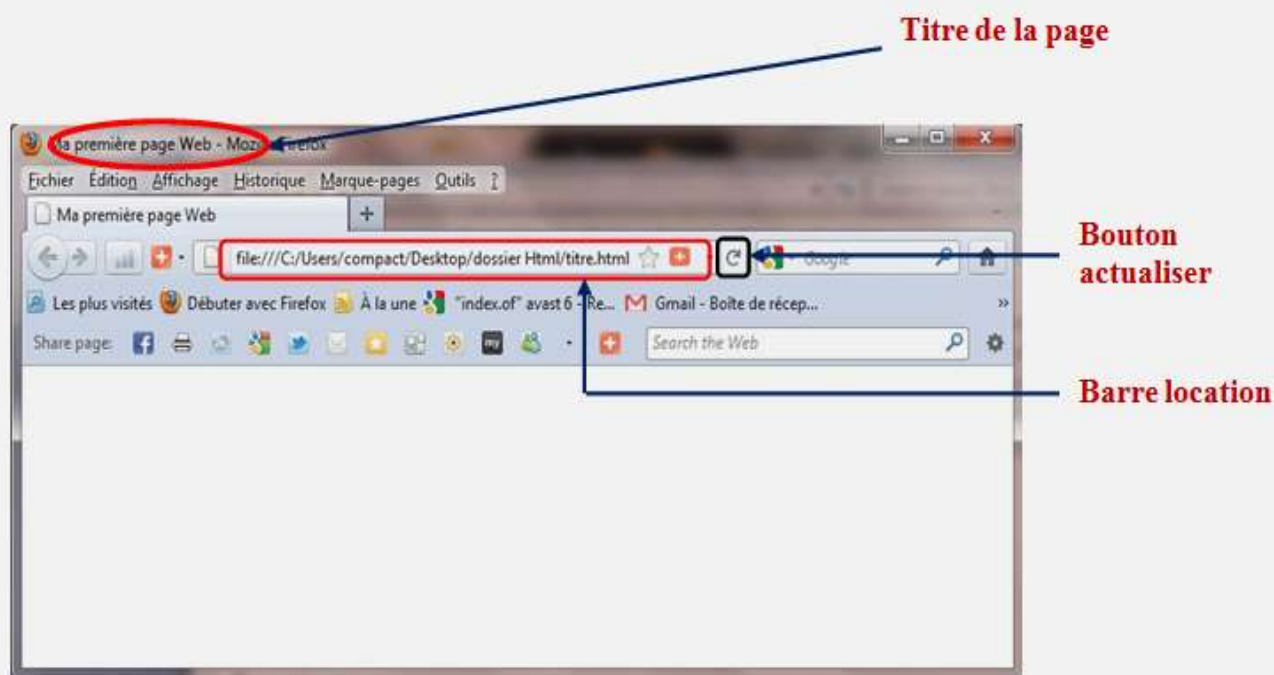
<HTML>
  <HEAD>
    <TITLE>Ma première page Web</TITLE>
  </HEAD>

  <BODY>

  </BODY>
</HTML>

```

- ⊕ Enregistrer le document avec l'extension .html ou .htm.
- ⊕ Ouvrir votre navigateur.
- ⊕ Afficher le document via le menu File/Open file...
- ⊕ Admirez votre premier document Html



Celui-ci est vide (et c'est normal) mais tout à fait opérationnel! Il faudra maintenant lui fournir votre information à l'intérieur des balises `<BODY></BODY>`. Remarquez que votre "TITLE" est présent dans la fenêtre de votre navigateur.

Pour vos éventuelles modifications, il n'est pas nécessaire de rouvrir à chaque fois le navigateur.

- ▶ Retourner dans l'éditeur de texte (sans fermer le navigateur).
- ▶ Modifier les codes Html.
- ▶ Enregistrer le fichier.
- ▶ Utiliser la commande **Actualiser (Reload)** du navigateur ou si celui-ci est paresseux cliquer dans la barre "Location" et faire "Enter".

Tout document Html contiendra en majorité du texte. Son enrichissement se fait à travers des balises qui permettent d'adapter le contenu en fonction de ces dernières. Voyons comment l'agrémenter par quelques balises élémentaires.

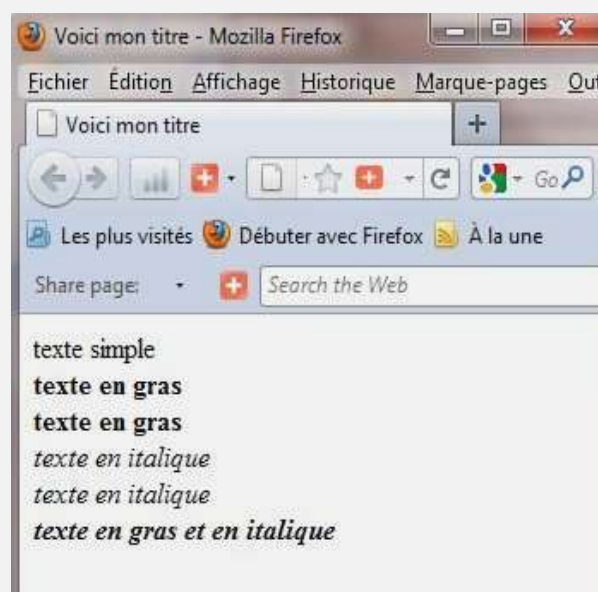
Gras [Bold]	<code>...</code> ou <code>...</code>
Italique [Italic]	<code><I>...</I></code> ou <code>...</code>
Taille de caractère [Font size]	<code>...</code>
Couleur de caractère [Font color]	<code> </code>
A la ligne [Line break]	<code>
</code>
Commentaires [Comments]	<code><!-- *** --></code>
Centrage [Center]	<code><CENTER></CENTER></code>

Ouvrir l'éditeur de texte, saisir le texte suivant et l'enregistrer sous format Html. Puis, ouvrir votre document html dans votre navigateur pour admirer votre travail:

```

<HTML>
<HEAD>
<TITLE>Voici mon titre</TITLE>
</HEAD>
<BODY>texte simple<BR>
<B>texte en gras</B><BR>
<STRONG>texte en gras</STRONG>
<BR><I>texte en italique</I><BR>
<EM>texte en italique</EM><BR>
<B><I>texte en gras et en italique</I>
</B><BR>
<BODY>
</HTML>

```



III- Les balises

La description d'un document HTML passe par l'utilisation de **BALISES** (ou "**TAGS**" en anglais). Une balise est délimitée par les signes "<" et ">" entre lesquels figure le nom de la balise. Par exemple, la balise de retour à la ligne est **
. La plupart du temps, on utilise une balise de début et une balise de fin, qui définissent les propriétés du contenu de la dite balise. Une balise peut avoir des **attributs qui adaptent le contenu de ladite balise. **<DIV align=center>...</DIV>** (**align** ici est attribut et **center** sa valeur)



Quand vous écrirez les balises de votre page Html, il faudra garder à l'esprit :

- ✚ qu'une balise marque une action pour le navigateur (ce qu'il doit faire...).
- ✚ que les attributs précisent les modalités de cette action (comment il doit le faire...).
- ✚ que toutes les balises ne sont pas fermantes (
, , etc.)

I. Liste de quelques balises HTML

La présente liste de balises n'est pas exhaustive mais présente une panoplie de balises généralement utilisées par les concepteurs de pages web.

SYNTAXES	EXPLICATIONS
<HTML> ... </HTML>	Indique que le document est écrit en Html
<HEAD> ... </HEAD>	En-tête du document Html
<TITLE> ... </TITLE>	Titre du document Html qui sera affiché dans la barre de titre de votre navigateur internet.
<BODY> ... </BODY>	Corps du document Html qui sera affiché dans la fenêtre du navigateur.

 	Force un saut de ligne
 	Met un texte en gras le texte
<I> ... </I> ... 	Met un texte en italique le texte
<U> ... </U>	Souligne un texte
 ... 	Fixe une taille de police pour un texte
 ... 	Fixe une couleur pour un texte
<CENTER> ... </CENTER>	Centre un texte
<!-- *** -->	Insère un commentaire qui n'apparaît évidemment pas dans le navigateur
<Hn> ... </Hn>	Insère des en-têtes qui servent à diviser le texte en sections
<P> ... </P>	Saut de ligne, insère une ligne vierge et commence un paragraphe
<PRE> ... </PRE>	Affiche un texte préformaté et prend en compte les espaces et sauts de ligne définis dans l'éditeur de texte.
_{...}	Met un texte en indice
^{...}	Met un texte en exposant
<DIV align= ?> ... </DIV>	Aligne un texte à gauche / centre / droite
 ... 	Affiche le texte sous forme d'une liste non-ordonnée.
 ... 	Affiche le texte sous forme d'une liste ordonnée.
	Insère un élément de la liste
<NOBR> ... </NOBR>	Met du texte sans retour à la ligne
<WBR> ... </WBR>	Force un retour à la ligne dans un texte encadré par <NOBR>... </NOBR>
<HR>	Insère une ligne horizontale de séparation.
<BLINK> ... </BLINK>	Affiche un texte clignotant.
<ADDRESS> ... </ADDRESS>	Indiquer une adresse (généralement en fin de document).
	Affiche l'image qui se trouve à l'adresse...
 ... 	Insère un lien pour se transporter d'une page web à l'autre.
Les ancrés: 	Insère un point d'ancrage dans une page web Insère un lien vers une ancre dans la même page Lien vers une ancre dans une autre page

<BODY BGCOLOR="#\$\$\$\$\$">	Insère une couleur d'arrière-plan
<BODY BACKGROUND="Adresse">	Insère une texture d'arrière-plan se trouvant à l'adresse... dans un dossier de votre ordinateur
<TABLE> ... </TABLE>	Définit un tableau [Table]
<TR> ... </TR>	Définit une ligne [Table Row]
<TD> ... </TD>	Définit une cellule [Table Data]
<TD width=?> en pixels ou % <TD rowspan=?> <TD colspan=?>	Largeur d'une cellule Fusion de lignes Fusion de colonnes
<FRAMESET> ... </FRAMESET> <FRAMESET ROWS=" ... "> <FRAMESET COLS=" ... ">	Début et fin d'une zone avec des fenêtres Fenêtres horizontales Fenêtres verticales
<NOFRAMES> ... </NOFRAMES>	Indique le texte que doivent afficher les navigateurs incapables de gérer les frames
<CODE> texte </CODE>	Utilise une police à chasse fixe (encombrement caractères constant).
<SAMP> ... </SAMP>	séquence de caractères littéraux.
<KBD> ... </KBD>	Met en évidence une saisie d'utilisateur.
<VAR> ... </VAR>	Indique le nom d'une variable.
<DFN> ... </DFN>	Met en évidence la 1ère utilisation d'un terme.
<CITE> ... </CITE>	Met en évidence une citation (généralement en italique).
<ADDRESS> ... </ADDRESS>	Indique l'auteur d'un document ainsi que le moyen de le contacter ou donne l'adresse du document
<BLOCKQUOTE> ... </BLOCKQUOTE>	Introduit une citation avec un léger retrait à gauche et/ou à droite.
<STRIKE> ... </STRIKE>	Barre un texte.
<LISTING> code informatique </LISTING>	Affiche un code informatique
<BIG> ... </BIG>	Écrit un texte en plus gros caractères que les caractères en cours
<SMALL> ... </SMALL>	Écrit un texte en plus petits caractères que les caractères en cours

2. Utilisation des balises

Nous allons aborder cette partie en mettant en pratique quelques travaux pratiques que nous appelons ici activités. Il sera premièrement question d'utiliser les balises de division et la mise en valeur d'un texte. Deuxièmement, nous mettrons en œuvre les balises des listes.



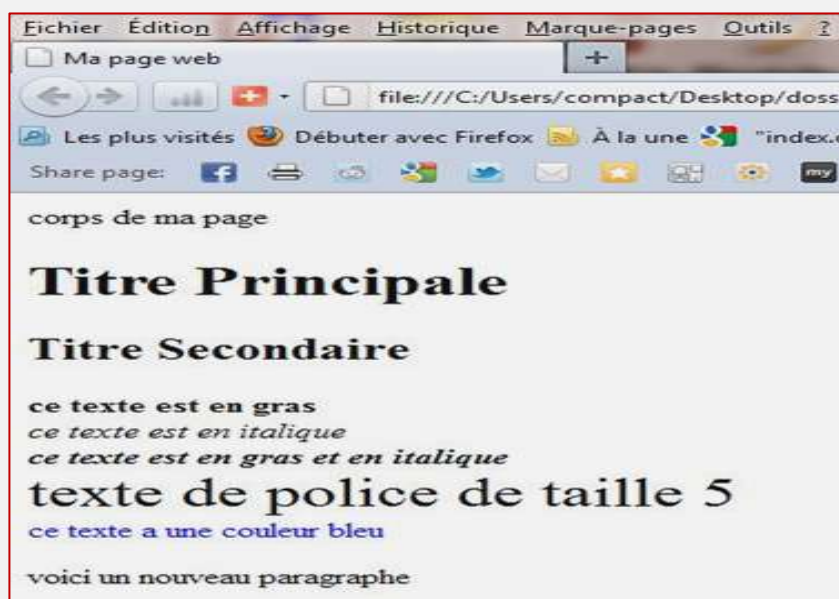
Activité 1 : utiliser les balises de division et de mise en valeur d'un texte.

Voici le script d'une page html utilisant les balises de division et de mise en valeur d'un texte et voila ci-après sa page web ouvert dans un navigateur.

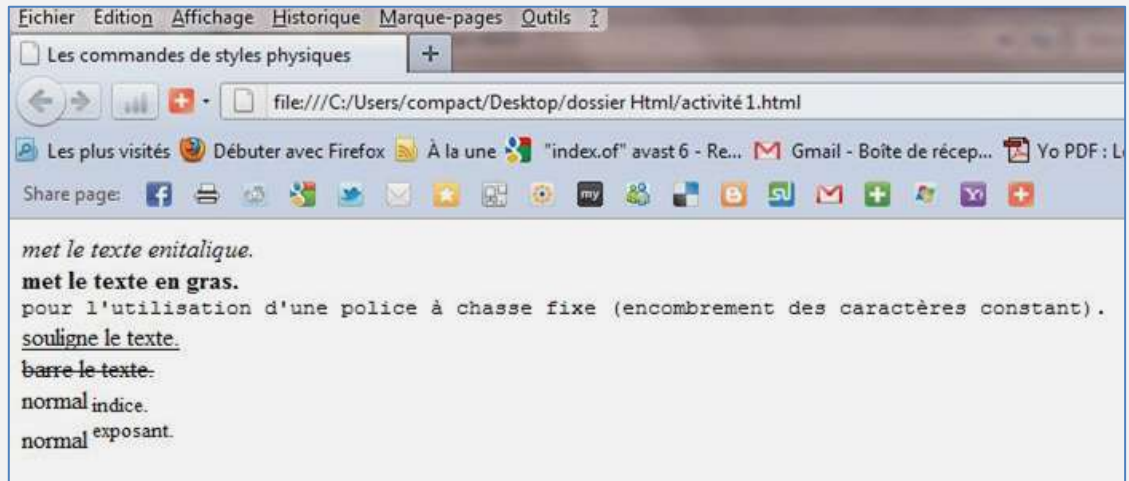
```

1  <HTML>
2  <HEAD>
3      <TITLE> Ma page web</TITLE>
4  </HEAD>
5  <BODY> corps de ma page<BR>
6      <H1> Titre Principale </H1>
7      <H2> Titre Secondaire </H2>
8      <B> ce texte est en gras</B><BR>
9      <I>ce texte est en italique</I><BR>
10     <B><I>ce texte est en gras et en italique</I></B><BR>
11     <FONT SIZE=6>texte de police de taille 5</FONT><BR>
12     <FONT COLOR="#0000FF">ce texte a une couleur bleu</FONT>
13     <P> voici un nouveau paragraphe </P>
14     <!--C'est fini-->
15 </BODY>
16 </HTML>

```

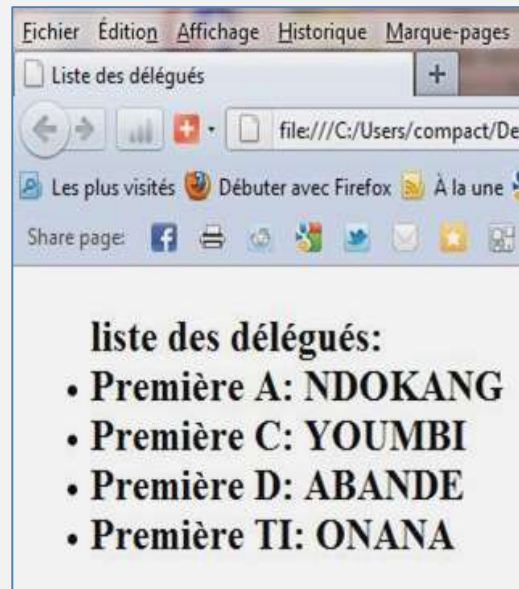
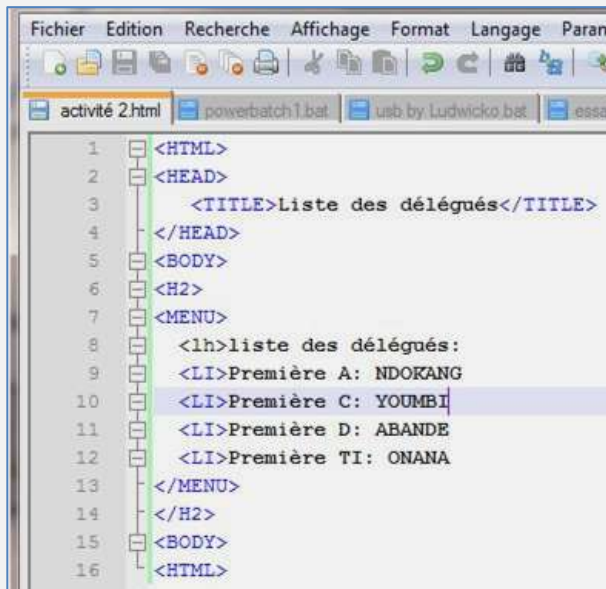


APPLICATION : La page ci-après est un document Html ouvert dans un navigateur. En s'inspirant de l'activité précédente, écrire le script Html de cette page web.

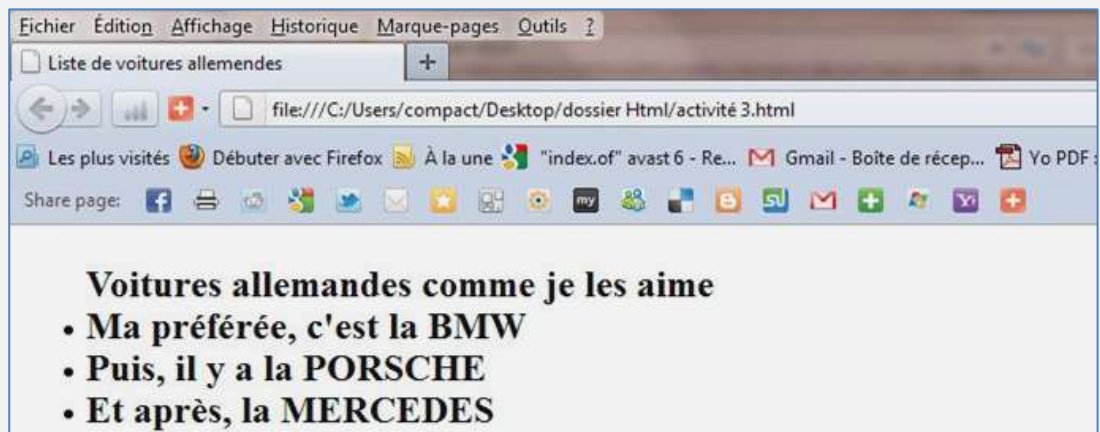


Activité 2 : utiliser les balises de listes.

Voici le script d'une page html utilisant les balises de listes et voilà ci-après sa page web ouverte dans un navigateur.



APPLICATION : Voici la page d'un document Html ouvert dans un navigateur. En s'inspirant de l'activité précédente, écrire le script Html de cette page web.



Les caractères spéciaux

Les caractères spéciaux en Html ne s'intègrent pas régulièrement comme on le pense. On utilise les séquences d'échappement suivantes pour les remplacer dans le cas par exemple des accents pris par la voyelle A / a.

&Aacute = Á	&Aring = Å	&aacute = á	&agrave = à
&Acirc = Â	&Atilde = Ã	&acirc = â	&atilde = ã
&Agrave = À	&Auml = Ä	&aring = å	&auml = ä

Ces caractères spéciaux s'appliquent sur les autres lettres ayant en général des accents comme E, e, U, u, I, i, O, o, N, n, Y ou encore y.

Autres caractères spéciaux

&amp; = &	&AElig = Æ	&ETH = Ð	&THORN = Þ	&copy = ©
&quot; = "	&aelig = æ	&eth = ð	&thorn = þ	&reg = ®
&ccedil = ç	&Ccedil = Ç	&Oslash = Ø	&szlig = ß	&oslash = ø



Résumé

HTML est le langage universel utilisé pour communiquer sur le Web.

Un **site web** est un ensemble de pages web liés par des liens permettant d'aller d'une page à l'autre.

Une **page web** est la description d'un code Html. Le langage Html est à la base de la création des sites web.

La description d'un document Html passe par l'utilisation de **BALISES**. Une balise est délimitée par les signes "<" et ">" entre lesquels figure le nom de la balise. Elle peut avoir des **attributs** qui adaptent le contenu de ladite balise.

Voici un exemple de code Html de base :

```
<HTML>
|
|   <HEAD>
|   |   <TITLE>Voici mon titre</TITLE>
|   </HEAD>
|   <BODY>voici le corps de ma page web<BR>
|   |   <B>texte en gras</B>
|   |   <I>texte en italique</I><BR>
|   |   <B><I>texte en gras et en italique</I></B>
|   </BODY>
</HTML>
```

L'utilisation des caractères spéciaux dans le langage Html se fait via l'usage des séquences d'échappement présentées à la fin du chapitre.

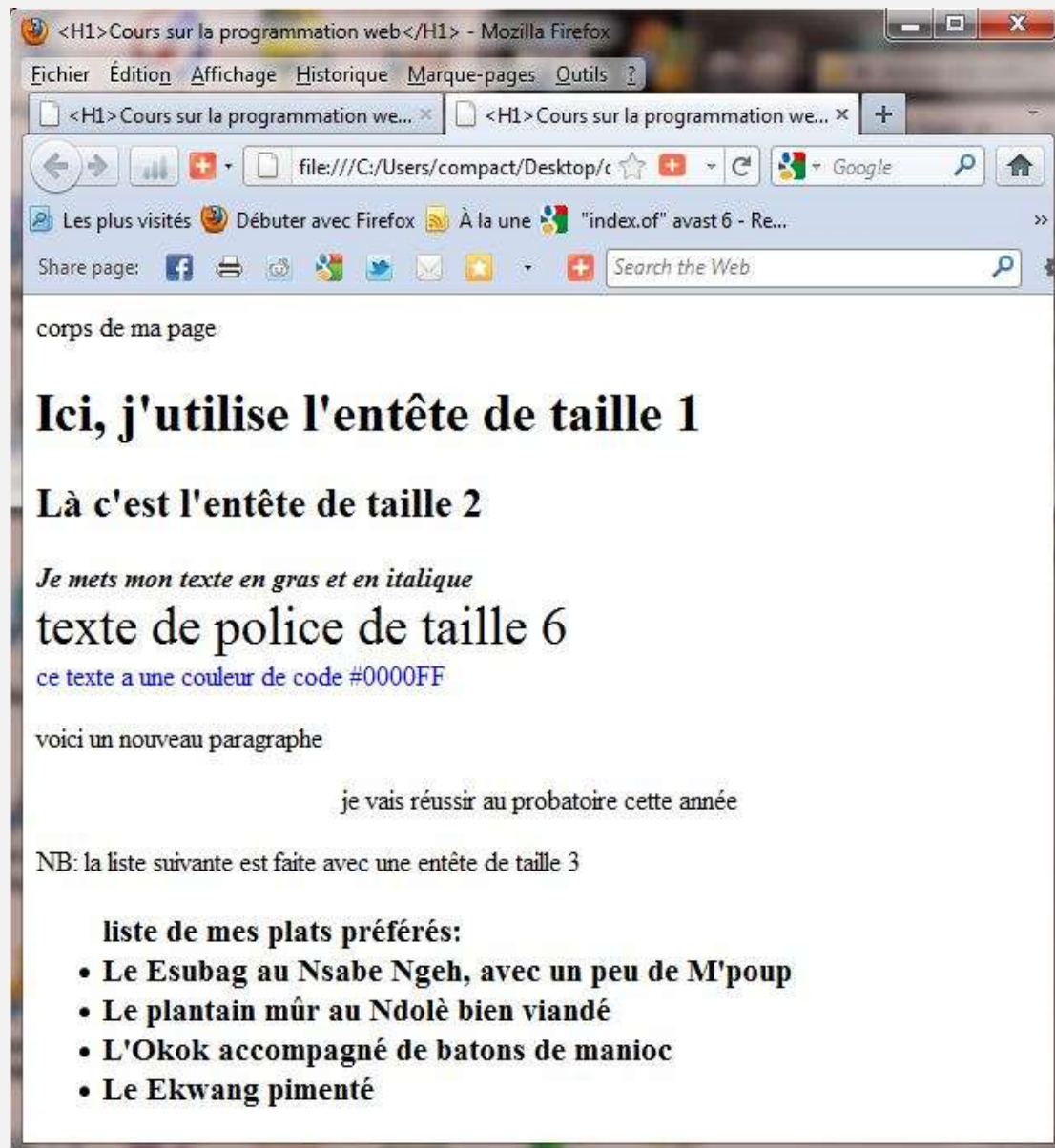


Exercice 1 : questions de cours

1. Définir les termes suivants : html, page web, site web, balise.
2. Donner la structure de base d'un code html en expliquant les fonctions jouées par chaque élément du code.
3. Donner au moins trois avantages liés à l'utilisation du langage html dans la programmation des pages web.

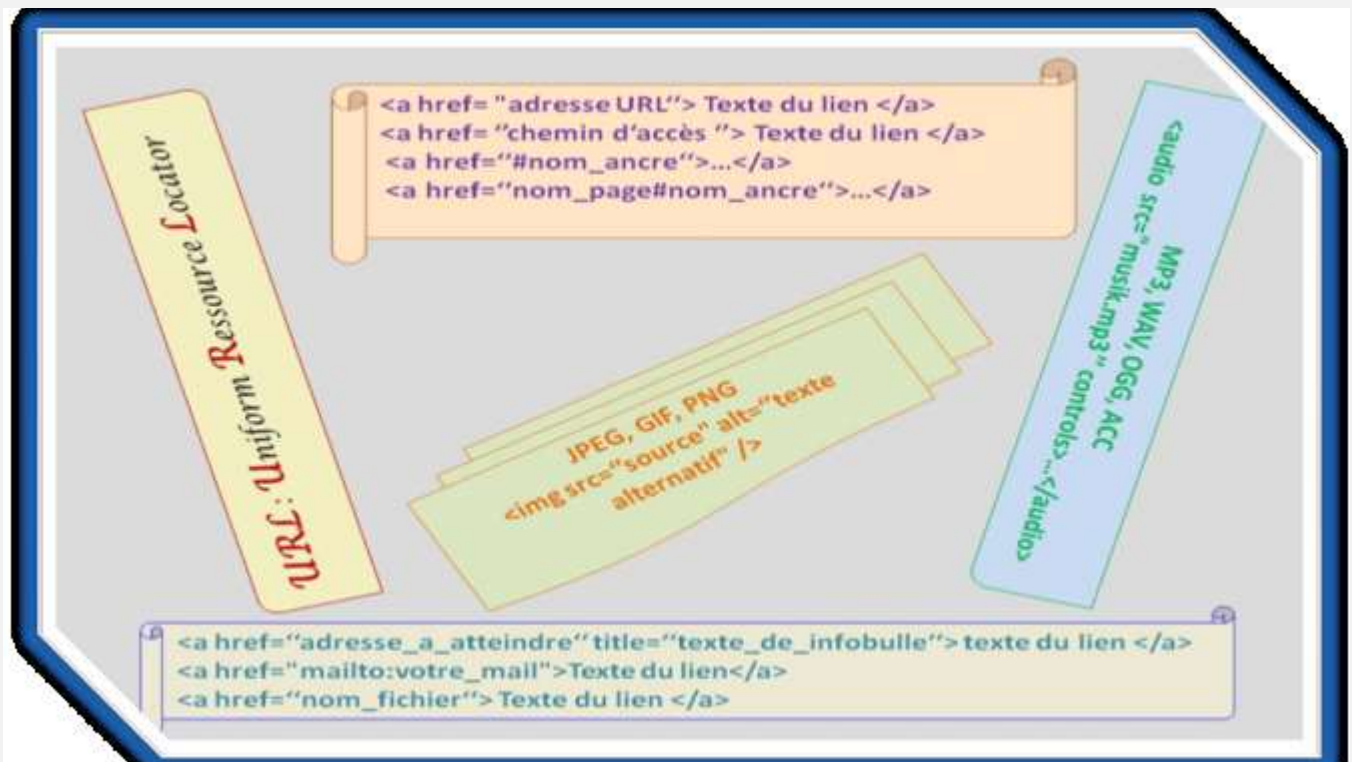
Exercice 2 : consolidation des connaissances

En vous appuyant sur les différentes activités vues dans le cours et sur la liste des balises, écrire le code de cette page web dans un éditeur de texte et puis, le tester.



Chapitre 10

INSERTION DE LIENS, IMAGES ET SONS DANS UNE PAGE WEB



SOMMAIRE



- I- INSERTION DES LIENS
- II- INSERTION DES IMAGES
- III- INSERTION DES SONS AUDIO



Problème

Vous faites partie du groupe chargé de la réalisation d'un site web pour votre lycée. Après avoir créé plusieurs pages pour votre site, vous devez trouver un moyen qui permettra au visiteur de votre site de :

- se déplacer d'une page à une autre ;
- se déplacer plus rapidement à l'intérieur d'une page longue ;
- se diriger vers d'autres sites ;
- visualiser des images et écouter des sons.

Comment allez-vous procéder pour le faire ?



Pré-requis

- Connaître les usages d'un éditeur de texte et d'un navigateur ;
- Connaître la structure de base d'un document HTML ;
- Savoir utiliser les balises ;
- Produire et tester un document HTML minimum.



Objectifs

- Insérer un lien vers une autre page, un autre site et une ancre ;
- Insérer une image et un son ;
- Insérer une bulle d'aide ;
- Insérer un lien pour envoyer un mail ;

INTRODUCTION

Le chapitre précédent a permis la création d'un document HTML minimum. Un **site web** est constitué de plusieurs pages et il est indispensable de pouvoir naviguer à travers toutes ces pages. HTML est un **langage hypertexte** qui permet en cliquant sur un mot, une expression (généralement souligné) ou une image, de se transporter vers un autre endroit du document, vers un autre fichier Html situé sur l'ordinateur local, vers un autre ordinateur situé sur le Web. Les liens hypertexte permettent de surfer de page en page et constituent l'essence même des documents HTML. Une **image** peut également constituer un lien. Dans ce chapitre, nous présenterons comment insérer un lien, une image, un son audio dans une page web.

I. INSERTION DES LIENS

Un **lien** est un texte ou une image sur lequel on clique pour se rendre sur une autre page. Pour faire un lien, la balise utilisée est très simple: `<a>.....</a>`. Il faut cependant lui ajouter l'attribut **href** pour indiquer vers quelle page on souhaite se rendre. Il est facile de reconnaître les liens sur une page : ils sont écrits d'une façon différente (par défaut en bleu souligné) et un curseur en forme de main apparaît lorsqu'on pointe dessus.

1. Un lien vers un autre site

a. Notion d'adresse URL

URL (Uniform Resource Locator) c'est une adresse web qui permet de mémoriser plus facilement une adresse IP. En d'autres termes, sur Internet, les documents sont ainsi

repérés par une adresse unique, appelée **URL**, permettant de localiser une ressource sur n'importe quel serveur du réseau internet. Une **URL** se présente sous la forme suivante :

<http://www.nomdusite.TLD>

EXEMPLE ILLUSTRATIF :

<http://www.informatiquepourtous.com>

<http://www.commentcamarche.com/www/www-intro.php3>

- ⓐ **http://** indique que nous souhaitons naviguer sur le web au moyen du **protocole HTTP**, le protocole utilisé par défaut pour naviguer entre les pages web.
- ⓑ **www.commentcamarche.com** cette adresse est appelée **nom de domaine** et correspond à **l'adresse du serveur** qui héberge les pages web. Par convention les serveurs web possèdent un nom commençant par **www**, afin de bien symboliser qu'il s'agit de serveurs dédiés à l'usage du web.
- ⓒ Le **TLD (Top Level Domain)** sont des noms de domaines de niveau supérieur proposant une classification des sites web selon leurs secteurs d'activités. On a par exemple .com, .edu, .cm, .org, .net ...
- ⓓ **/www/www-intro.php3** permet d'indiquer la localisation du document sur la machine. Dans le cas présent il s'agit du fichier **www-intro.php3** situé dans le répertoire **www**.

b. Syntaxe d'insertion du lien

La syntaxe d'insertion d'un lien vers un autre site est la suivante :

`<a href= "adresse URL"> Texte du lien </a>`

Où **adresse URL** désigne l'adresse URL du site vers lequel on souhaite diriger le visiteur et **texte du lien**, le texte qui sera affiché pour représenter le lien.



Lancer un éditeur de texte et entrer le code HTML suivant. Enregistrer le fichier comme une page web et tester le.

```
<html>
<head>
<title> mon premier lien </title>
</head>
<body>
<p>Bonjour! Souhaitez-vous
visiter le
<a href="http://www.dite-
ens.cm"> Site du DITE </a> ?
<br/> C'est un bon site ! </p>
</body> </html>
```



2. Lien vers une autre page du site

L'organisation classique, et plus que conseillée, d'un site Web consiste à regrouper l'ensemble des éléments de celui-ci (fichiers html, images, ...) dans un même répertoire. Mais ceci n'est pas toujours le cas. La syntaxe d'insertion d'un lien vers une autre page du site est la suivante :

`<a href= "chemin d'accès "> Texte du lien </a>`

Le **chemin d'accès** dépend selon que les pages se trouvent dans le même dossier ou non.

a. Les pages sont dans le même dossier

Le chemin d'accès est simplement le nom de la page. `<a href="page.html"> texte </a>`

b. Une des pages est dans un sous dossier

Le chemin d'accès est `<a href="nom-sous-dossier/ page.html"> texte </a>`.

Si la page se trouve dans un autre dossier du sous dossier alors le chemin d'accès devient `<a href="nom-sous-dossier/nom-autre-dossier/page.html"> texte </a>`

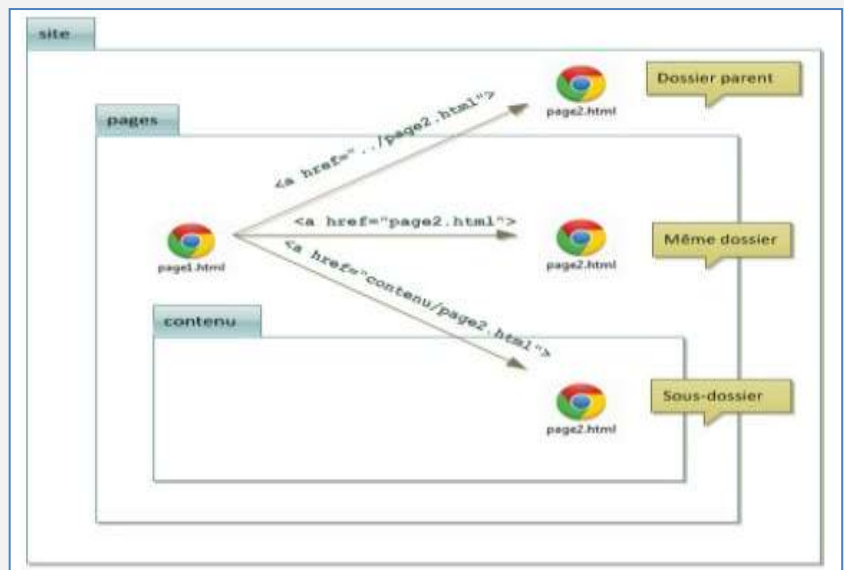
Il faut donc dans ce cas spécifier à chaque fois le **chemin d'accès absolu** au fichier HTML.

c. Une des pages est dans un dossier parent

Le chemin d'accès est `<a href=" ../ page.html"> texte </a>`.

ILLUSTRATION :

On dispose de deux pages `page1.html` et `page2.html`
On souhaite faire un lien de la `page1` vers la `page2`. Les différentes possibilités qu'on peut avoir sont schématisées ci-contre



3. Lien vers une ancre

Une ancre est une sorte de point de repère qu'on place dans des pages web étendues. En effet, si une page est très grande il peut être utile de faire un lien amenant plus bas dans la même page pour que le visiteur puisse passer directement à la partie qui l'intéresse.

Pour créer une ancre, il suffit de rajouter l'**attribut id** à une balise qui va alors servir de repère. Cet attribut permet de donner un nom à l'ancre ce qui servira ensuite à faire un lien vers cette ancre. On peut ajouter l'attribut id à n'importe quelle balise.

Exemple : si on insère l'ancre "physique" dans un titre comme suit

`<h2 id="physique"> introduction </h2>`

Il suffit au moment d'insérer le lien de procéder comme d'habitude, mais cette fois l'attribut **href** contiendra un **dièse (#)** suivi du nom de l'ancre. L'instruction permettant d'insérer le lien est donc la suivante `<a href="#physique">Aller vers introduction</a>`



Pour que le lien vers une ancre fonctionne, il faut que la page comporte suffisamment de texte pour que les barres de défilement se déplacent automatiquement.

4. Lien vers une ancre située dans une autre page

L'idée, c'est de faire un lien qui ouvre une nouvelle page et qui amène directement à une ancre située plus bas sur cette page. En pratique, il suffit de taper le **nom de la page**, suivi d'un **dièse (#)**, suivi du **nom de l'ancre**. `<a href="nom_page#nom_ancre">...</a>`

Exemple : on dispose de deux pages `page1.html` et `page2.html`. La page2 possède une ancre nommée **physique** comme dans l'exemple précédent. On se place à la page1 et on désire insérer un lien qui mènera directement à l'ancre physique de la page2. L'instruction est la suivante : `<a href="page2.html#physique"> aller vers introduction </a>`

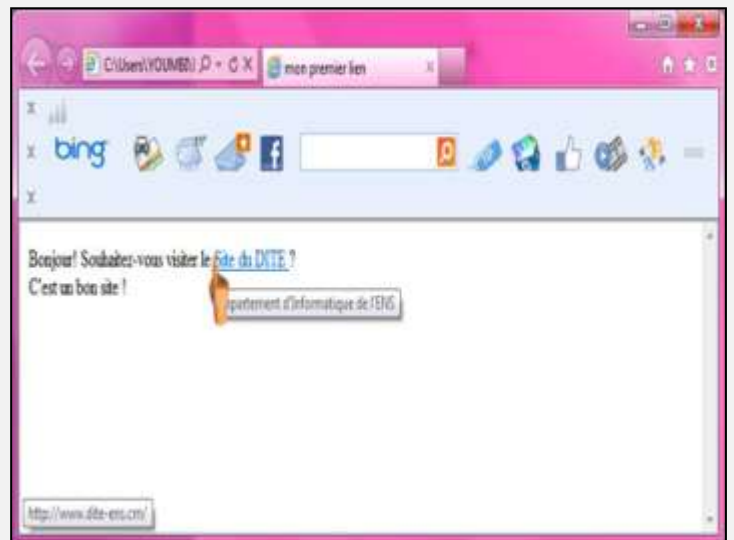
5. Lien qui affiche une infobulle au survol

On peut utiliser l'attribut **"title"** qui affiche une **bulle d'aide** lorsqu'on **pointe sur le lien**. La bulle d'aide peut être utile pour informer le visiteur avant même qu'il n'ait cliqué sur le lien. On a la syntaxe suivante :

`<a href="adresse_a_atteindre" title="texte_de_infobulle"> texte du lien </a>`

Exemple : reprenons-le code Html de l'activité précédente et ajoutons l'attribut **title** à la balise d'insertion du lien. On obtient l'instruction suivante :

`<a href="http://www.dite-ens.cm" title="département d'Informatique de l'ENS"> Site du DITE </a>`



6. Lien pour envoyer un mail

Si on désire que les visiteurs d'un site web puissent envoyer un mail au propriétaire, on doit utiliser des liens de **type "mailto"**. Rien ne change au niveau de la balise, on doit simplement modifier **la valeur de l'attribut href** comme ceci :

Exemple : on désire insérer un lien qui permet d'envoyer un mail à l'adresse suivante `youbich@yahoo.com`, l'instruction est la suivante :

`<a href="mailto:youbich@yahoo.com"> Envoyez un mail </a>`

7. Un lien pour télécharger un fichier

Pour insérer un lien qui permet au visiteur du site de télécharger un fichier, il faut procéder de la même façon que l'insertion d'un lien vers une page web, mais en indiquant le **nom du fichier** à télécharger et en précisant son **extension**. On a la syntaxe `<a href="nom_fichier"> Texte du lien </a>`. Il est à noter que le fichier doit se trouver dans le dossier du site web en question.

Exemple : soit à insérer un lien pour le téléchargement du fichier **footballeurs.doc**
`<a href="footballeurs.doc"> Télécharger la liste des footballeurs </a>`



Étape 1 : insertion d'un lien vers une autre page de son site

Créer deux pages web avec les codes ci-dessous et placer les dans un même dossier.

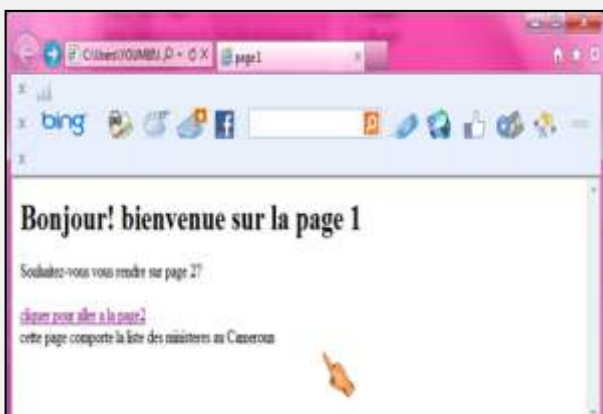
PAGE 1:

```
<html>
<head> <title> page 1
</title> </head>
<body>
<h1> Bonjour! bienvenue sur la
page1 </h1>
<p> Souhaitez-vous vous rendre sur
page2? </p>
<a href="page2.html"> cliquer pour aller
a la page2 </a> <br />
cette page comporte la liste des
ministeres au Cameroun </p>
</body>
```

PAGE 2:

```
<head>
<title> page 2 </title>
</head>
<body>
<h1> Bonjour! bienvenue sur la page
2 </h1>
<h2> liste des ministeres au
Cameroun </h2>
</body>
</html>
```

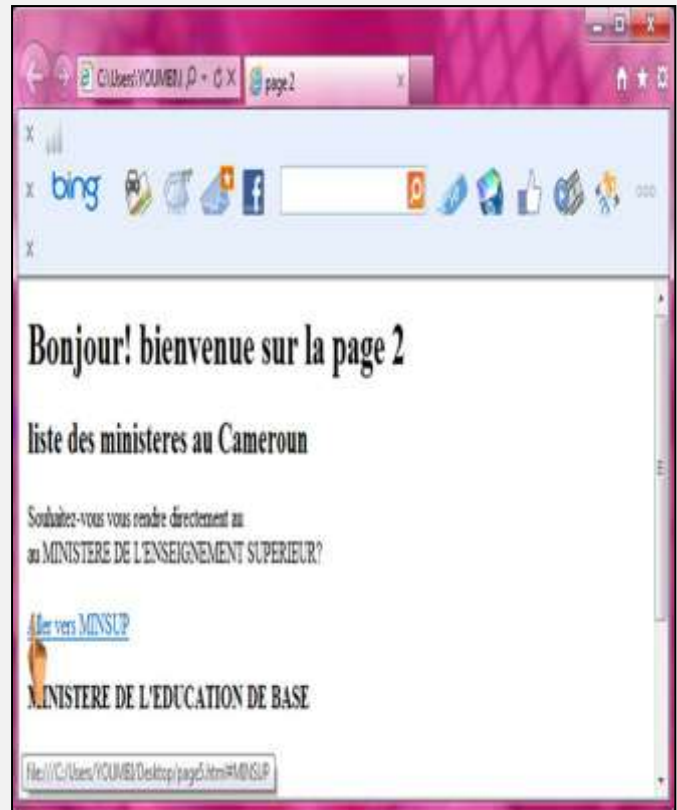
Les pages web se présentent de la façon suivante :



Étape 2 : lien vers une ancre située dans le même document

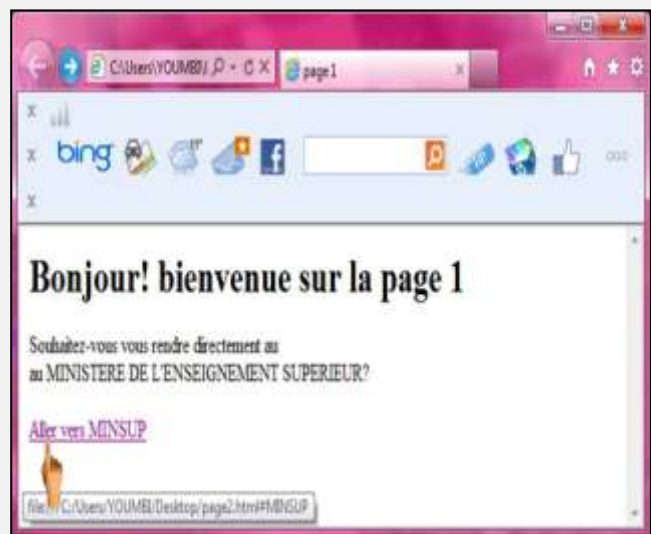
Modifions le code de la page2 ci-dessus de sorte que son contenu soit suffisamment dense afin de déplacer automatiquement les barres de défilement.

```
<html>
<head> <title> page 2 </title> </head>
<body>
<h1> Bonjour! bienvenue sur la page 2 </h1>
<h2> liste des ministeres au Cameroun </h2>
<p> Souhaitez-vous vous rendre directement
au <br> au MINISTERE DE
L'ENSEIGNEMENT SUPERIEUR? </p>
<a href="#MINSUP"> Aller vers
MINSUP </a>
<h3> MINISTERE DE L'EDUCATION
DE BASE </h3>
<p> En charge des <br>
<h3 id="MINSUP"> MINISTERE DE
L'ENSEIGNEMENT SUPERIEUR </h3>
<p> En charge des <br>
des Universites d'Etat <br>
des Universites privées <br> </p>
</body> </html>
```

**Étape 3 : Lien vers une ancre située dans une autre page**

Modifions le code de la page 1 de façon à y insérer un lien qui permette d'atteindre l'ancre MINSUP de la page 2 :

```
<html>
<head> <title> page 1 </title> </head>
<body>
<h1> Bonjour! bienvenue sur la page 1 </h1>
<p> Souhaitez-vous vous rendre
directement au <br> au MINISTERE DE
L'ENSEIGNEMENT SUPERIEUR?
</p>
<a href="page2.html#MINSUP"> Aller
vers MINSUP </a>
</body> </html>
```



II. INSERTION DES IMAGES

I. Les différents formats d'images

Une image peut être enregistrée dans plusieurs formats différents. Le poids (en Ko, voire en Mo) et la qualité de l'image sera plus ou moins nette selon le format choisi. Certains formats sont plus adaptés que d'autres selon l'image (photo, dessin, image animée...) et de même certains sont plus adaptés pour le net que d'autres. On distingue pour :

- ② Une photo : utilisez un JPEG.
- ② N'importe quel graphique avec peu de couleurs (moins de 256) : utilisez un PNG 8 bits, ou éventuellement un GIF.
- ② N'importe quel graphique avec beaucoup de couleurs : utilisez un PNG 24 bits.
- ② Une image animée : utilisez un GIF animé.

2. Insérer une image

La balise permettant d'insérer une image dans une page web est `<img/>`. C'est une balise non fermante c'est-à-dire qu'on n'a pas besoin de l'écrire en deux exemplaires comme la plupart des balises utilisées jusqu'ici. La balise doit être accompagnée de 2 attributs obligatoires :

- ② SRC : signifie "source". Il permet d'indiquer où se trouve l'image que l'on veut insérer.
- ② ALT : cela signifie "texte alternatif". On doit toujours indiquer un texte alternatif à l'image, c'est-à-dire un texte court qui décrit ce que contient l'image.

Exemple : soit à insérer l'image promo2010.jpg dans une page web, l'instruction est la suivante

```
<p>voici une photo des etudiants de la promotion arc en ciel <br />
 </p>
```

REMARQUE

1. Les images doivent se trouver à l'intérieur d'un paragraphe (`<p></p>`).
2. Éviter les caractères accentués, les espaces et les majuscules dans les noms de fichiers, dossiers et images.
3. Il faut toujours s'assurer que l'image est dans le même dossier ou dans un sous dossier du dossier qui contient les pages web

3. Ajouter une infobulle

L'attribut qui permet d'afficher une bulle d'aide est le même que pour les liens : il s'agit de `title`. Cet attribut est facultatif (contrairement à `alt`).

Exemple : soit à insérer l'image promo2010.jpg dans une page web, on a l'instruction :

```
<p>voici une photo des etudiants de la promotion arc en ciel <br />
 </p>
```

4. Insérer une image ou une miniature cliquable

Si l'image est très grosse, il est conseillé d'en afficher la miniature sur le site. Pour cela il faut au préalable miniaturiser l'image à partir de logiciels spécialisés. Ajouter ensuite un lien sur cette miniature permettant aux visiteurs de pouvoir afficher l'image en taille originale. Il faut donc deux images, l'une étant la miniature de l'autre.

Exemple : soit à insérer l'image `promo2010.jpg` dans une page web avec une miniature `promo2010_mini.jpg`, on a l'instruction :

```
<p>souhaitez vous voir l'image dans sa taille normale ? cliquez dessus <br />
<a href="promo2010.jpg"> </a> </p>
```



Étape 1 : Insérer une image avec infobulle

```
<html>
<head><title> image
</title> </head>
<body>
<h1> Bonjour! bienvenue sur la page
image </h1>
<p>voici une photo des etudiants de la
promotion arc en ciel <br />
 </p>
</body> </html>
```



Étape 2 : Insérer une image avec infobulle et miniature

```
<html>
<head><title> image
</title> </head>
<body>
<h1> Bonjour! bienvenue sur la page
image </h1>
<p>souhaitez vous voir l image dans sa
taille normale ? cliquez dessus <br />
<a href="promo2010.jpg"> 
</a> </p>
</body> </html>
```



III. INSERTION DE SON AUDIO

Pour diffuser de la **musique** ou n'importe quel son, il existe de nombreux formats. La plupart d'entre eux sont compressés ce qui permet de réduire leur poids. On distingue les formats **MP3**, **WAV**, **OGG**, **ACC** etc. la balise qui permet d'insérer un son audio est **<audio>...</audio>** Cette balise nécessite des attributs :

- Ⓜ **SRC** : il permet d'indiquer où se trouve le son audio que l'on veut insérer
- Ⓜ **controls** : pour ajouter les boutons "**Lecture**", "**Pause**" et la **barre de défilement**
- Ⓜ **width** : pour modifier la largeur de l'outil de lecture audio.
- Ⓜ **loop** : la musique sera jouée en boucle.
- Ⓜ **autoplay** : la musique sera jouée dès le chargement de la page.
- Ⓜ **preload** : indique si la musique peut être pré chargée dès le chargement de la page ou non.

Exemple:

```
<html>
<head><title> musique </title>
</head>
<body>
<hr> Bonjour! bienvenue sur la page
musique</hr>
<p> <audio src="musik.mp3"
controls> </audio> </p>
</body></html>
```



La balise **<audio>...</audio>** n'est pas reconnu pas tout les navigateurs. On peut ne pas obtenir le résultat escompté.



Compléments

On peut également insérer une **vidéo** dans une page web à l'aide de la nouvelle balise **<video>...</video>**. Elle n'est pas reconnue par tous les navigateurs. Elle possède également de nombreux attributs : **src**, **poster** (image à afficher à la place de la vidéo lorsque celle-ci n'est pas lancée), **controls**, **width**, **height**, **loop**, **autoplay**, **preload**. Ces attributs ont les mêmes fonctions que ceux de la balise audio.



Résumé

La construction d'un site web nécessite la création de nombreuses pages. Cette création se fait grâce au HTML et ce langage offre la possibilité de pouvoir naviguer d'une page à l'autre ou même d'un site à l'autre. Le HTML se sert de la balise `<a>...</a>` pour établir ces liens. Cette balise a un attribut obligatoire "href" pour indiquer vers quelle page on souhaite être amené.

Un lien est un texte ou une image sur lequel on clique pour se rendre sur une autre page ou à un autre endroit de la page.

On distingue plusieurs types de liens :

- ② Les liens vers un autre site avec comme syntaxe `<a href= "adresse URL"> Texte du lien </a>`
- ② Les liens vers une autre page du site `<a href= "chemin d'accès "> Texte du lien </a>`. le chemin d'accès dépend du fait que les pages se trouvent dans le même dossier ou non.
- ② Les liens vers une zone dans la même page avec la syntaxe suivante `<a href = "# nom_ancree"> Texte du lien</a>`
- ② Les Liens vers une zone précise dans une autre page ayant comme syntaxe `<a href="nom_page#nom_ancree"> Texte du lien </a>`
- ② Les liens affichant une infobulle : `<a href="nom_du_lien" title="texte_de_infobulle"> texte du lien </a>`
- ② Les liens pour envoyer un mail : `<a href="mailto:votre_mail">Texte du lien</a>`
- ② Les liens pour télécharger un fichier : `<a href="nom_fichier"> Texte du lien </a>`

On peut insérer d'autres types d'objet dans une page web tels que les images, les sons ou même les vidéos.

Pour insérer une image, on utilise la balise `<img />` avec deux attributs obligatoires SRC et ALT. La syntaxe est la suivante : `<img scr= "chemin d'accès" alt="texte alternatif" />`.

- On peut ajouter une infobulle à une image grâce à l'attribut `title="texte de infobulle"`.
- On peut également insérer une image miniature sur laquelle il faut cliquer pour voir l'image grandeur nature. Il faut pour cela utiliser les balises `<a>...</a>` et `<img />`.

Pour insérer un son audio, on utilise la balise `<audio>....</audio>`. Cette balise n'est pas reconnue par tous les navigateurs et elle dispose de nombreux attributs tels que `src, controls, width, height, loop, autoplay, preload`.



EXERCICE 1 :

1. Définir les termes liens, ancre.
2. Donner la balise qui permet d'insérer un lien. Citer deux attributs de cette balise et donner leurs fonctionnalités.
3. Donner la balise qui permet d'insérer une image dans un page web. Citer trois attributs de cette balise et donner leurs fonctionnalités.
4. Donner la balise qui permet d'insérer un son audio dans un page web. Citer trois attributs de cette balise et donner leurs fonctionnalités.
5. Peut-on insérer une vidéo dans une page web ? si oui à l'aide de quelle balise ?

EXERCICE 2 :

Que font les instructions suivantes :

1. `<img scr= "fleur.png" alt="une rose" />`
2. `<a href="mailto:nancy.gmail.com"> envoyer moi un message </a>`
3. `<a href="cameroun.pdf"> télécharger l'histoire du Cameroun </a>`
4. `<a href="http://www.yahoo.fr"> yahoo france </a>`
5. ``

EXERCICE 3 :

- 1) Votre page "source.html" se trouve dans le répertoire "c:\site\source.html", et ma page "cible.html" se trouve dans "c:\site\documents\travaux\cible.html". Lequel de ces liens est le bon ?
 - (i) `<a href="documents/travaux/cible.html">`
 - (ii) `<a href="site/documents/travaux/cible.html">`
 - (iii) `<a href="cible.html">`
2. Votre page "source.html" se trouve dans le répertoire "c:\site\source.html" et je fais un lien vers "../cible.html". Où se trouve cible.html ?
 - (i) c:\dossier\cible.html
 - (ii) c:\..\cible.html
 - (iii) c:\cible.html

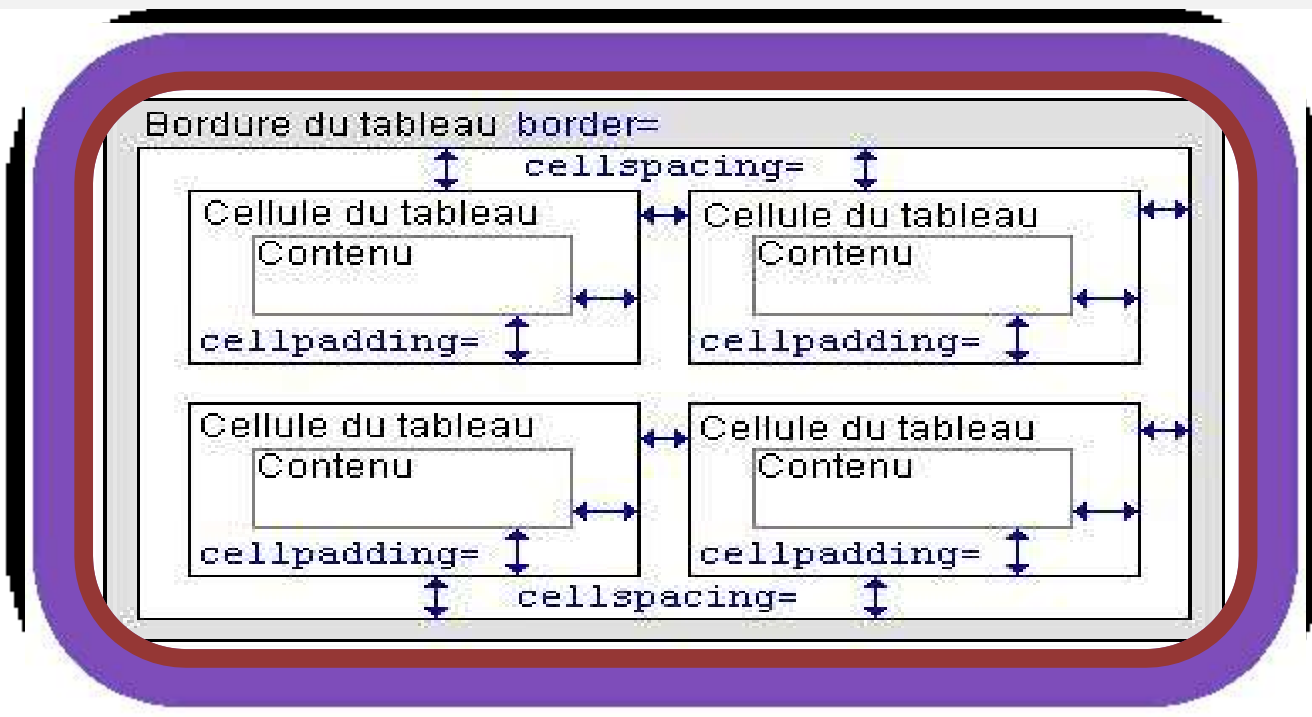
EXERCICE 4 :

Créer deux pages web. Nommer la première mathematique.html et la seconde physique.html.

- ⊗ Dans la page mathematique.html insérer une ancre qui permettra à partir du début du document de s'y rendre directement.
- ⊗ Dans la page physique.html insérer un lien qui mène directement à l'ancre du document mathematique.html.
- ⊗ Dans la page mathematique.html, insérer un lien vers la page physique.html.
- ⊗ Insérer une image de votre choix dans mathématique.html. Ajouter une infobulle à cette image
- ⊗ Insérer un son audio dans physique.html.
- ⊗ Dans physique.html, insérer une image cliquable (en miniature)

Chapitre 11

INSERTION DES TABLEAUX ET DES FRAMES DANS UNE PAGE WEB



SOMMAIRE



I. LES TABLEAUX

II. LES FRAMES



Problème

Les connaissances acquises jusqu'à lors vous permettent de créer des pages web. Votre enseignant de mathématiques aimerait avoir dans le site de l'école une page bien organisée dans laquelle on a plusieurs zones ou cadres pour son cours de statistique. Un de ces cadres contiendra l'énoncé du cours de statistique, un autre contiendra un exercice d'application comportant des tableaux de données, un troisième contiendra la solution de l'exercice ; on vous laissera le soin d'ajouter des publicités dans les autres cadres. Comment faudra-t-il procéder pour mettre sur pied cette page ?



Pré-requis

- Connaître la structure d'un code html
- Connaître le rôle d'une balise et comment utiliser ses attributs
- Être capable d'écrire un code html



Objectifs

- Insérer un tableau dans un code html
- Utiliser les balises et leurs attributs pour la mise en forme d'un tableau
- Découper une page en zones d'affichage des contenus

INTRODUCTION

La programmation web est un domaine de l'informatique qui permet de produire des pages web interconnectées. Ces pages ont une structure précise pour faciliter la navigation par les internautes. Le présent chapitre présente quelques outils permettant d'organiser les fenêtres. Dans un premier temps nous allons présenter **les tableaux** et comment les configurer, dans un second temps nous allons présenter **les cadres ou frames**.

I. LES TABLEAUX

Les tableaux ont été créés avant tout pour la représentation de données. Son usage est très fréquent en html. Ils servent non seulement à aligner des chiffres mais surtout à placer des éléments à l'emplacement que vous souhaitez. Vous pouvez définir des tableaux en HTML pour y présenter des données ou bien pour rendre plus attractive la présentation du texte et des graphiques à l'écran. Dans cette section, nous allons présenter l'utilisation des tableaux tout en listant de façon non exhaustive les balises et leurs attributs utilisés dans la manipulation des tableaux.

I. Définition des tableaux

a) Balises

Un **tableau** est un ensemble de lignes et de colonnes. À l'intersection de ces dernières se trouve **une cellule** du tableau dont le contenu peut être constitué aussi bien d'éléments de type bloc (titres, paragraphes, divisions, listes), d'éléments en ligne tels que des images, ou encore d'éléments de formulaire.

Les balises de création d'un tableau en HTML peuvent être illustrées par le schéma suivant :

<table>				
<tr>	<th>	<th>	<th>	</tr>
	</th>	</th>	</th>	
<tr>	<td>	<td>	<td>	</tr>
	</td>	</td>	</td>	
<tr>	<td>	<td>	<td>	</tr>
	</td>	</td>	</td>	
				</table>

<table> introduit un tableau (*table = tableau*). Si le tableau doit être quadrillé, vous devez ajouter l'attribut **border** dans la balise **<table>** et lui affecter une valeur supérieure à zéro (**0**). La valeur mentionnée est alors l'épaisseur du quadrillage en pixels. Pour créer un tableau sans bordure ni quadrillage visible omettez l'attribut **border** ou écrire **border="0"**. Il faut toujours fermer cette balise par la balise **</table>**

<tr> introduit une nouvelle ligne dans le tableau (*tr = table row = ligne de tableau*). À la suite sont définies les cellules de la ligne correspondante. À la fin d'une ligne de tableau est inscrit la balise de fermeture **</tr>**.

Un tableau peut contenir des cellules d'entête et des cellules de données normales. Le texte dans les cellules d'entête est mis en valeur (la plupart du temps en gras et centré).

<th> définit une cellule d'entête, **<td>** une cellule normale de données (*th = table header = entête de tableau, td = table data = données de tableau*). Le contenu d'une cellule est noté à chaque fois après la balise ouvrante. Bien que les balises de fermeture respectifs **</th>** ou bien **</td>** soient facultatifs, il est vivement recommandé de toujours les noter.



Si dans une ligne vous ne voulez pas entrer de données pour une colonne notez simplement **<td></td>**. Cependant certains navigateurs représenteront la cellule comme inexistante. Pour cette raison essayez donc aussi la suite de commande **<td> </td>** pour des cellules vides de tableau.

La balise **<colgroup>** permet de définir un groupe de colonnes. La balise **<col>** (n'est pas obligatoirement fermante) permet de définir des propriétés sur les colonnes à l'aide de l'attribut **width** qui donne la largeur de la colonne.

b) Exemples

Exemple1 : code html pour la création d'un tableau ayant pour épaisseur de bordure 1, de largeurs de colonne respectivement 80,100 et 320 pixels

```
<body>
  <table border="1">
    <colgroup>
      <col width="80">
      <col width="100">
      <col width="320">
    </colgroup>
    <tr>
      <th>Entête 1</th>
      <th>Entête 2</th>
      <th>Entête 3</th>
    </tr>
    <tr>
      <td>Ouest</td>
      <td>Nord ouest</td>
      <td>Sud ouest</td>
    </tr>
    <tr>
      <td>est</td>
      <td>Sud</td>
      <td>Littoral</td>
    </tr>
  </table>
</body>
```

Le résultat est donné comme suit

Entête 1	Entête 2	Entête 3
Ouest	Nord ouest	Sud ouest
est	Sud	Littoral

Exemple2 : code html pour la création d'un tableau ayant pour épaisseur de bordure 4 et ayant une image dans une cellule.

```
<body>
  <table border="4">
    <tr>
      <th>Entête 1</th>
      <th>Entête 2</th>
      <th>Entête 3</th>
    </tr>
    <tr>
      <td><p><img src=
        "sourisDansante.gif" width="68" height=
        "83" border="4" alt="petit rat"></p></td>
      <td>Nord ouest</td>
      <td>Sud ouest</td>
    </tr>
    <tr>
      <td>est</td>
      <td>Sud</td>
      <td>Littoral</td>
    </tr>
  </table>
</body>
```

Le résultat obtenu est le suivant :

Entête 1	Entête 2	Entête 3
	Nord ouest	Sud ouest
est	Sud	Littoral

2. Définir l'entête, le corps et le pied d'un tableau

a) Balises

Tout comme les têtes et pieds de page d'un document Word, les tableaux dans HTML peuvent être configurés pour avoir des pieds et têtes de tableau. Nous allons présenter les différentes balises définissant ces éléments.

<thead> permet d'introduire l'entête du tableau avec (*thead = table head = entête de tableau*). À la suite on peut insérer une ou plusieurs cellules à l'aide de la balise **<th>...</th>**. La clôture de l'entête du tableau se fait avec **</thead>**.

Le pied du tableau s'introduit avec **<tfoot>** (*tfoot = table foot = pied de tableau*). À la suite on peut insérer une ou plusieurs cellules à l'aide de la balise **<td>...</td>**. La clôture du pied du tableau se fait avec **</tfoot>**.

Le corps du tableau s'introduit avec **<tbody>** (*tbody = table body = corps de tableau*). On peut insérer une ou plusieurs lignes dans le corps du tableau. Chaque ligne est introduite par la balise **<tr>...</tr>**. À l'intérieur de la balise définissant une ligne, on insère chaque cellule à l'aide de la balise **<td>...</td>**. La clôture du pied du tableau se fait avec **</tbody>**.

b) Exemples

Exemple : code html pour la création d'un tableau ayant des pieds et têtes de page.

```
1 <table border="1">
2   <thead>
3     <tr>
4       <th>entete de tableau 1</th>
5       <th>entete de tableau 2</th>
6       <th>entete de tableau 3</th>
7     </tr>
8   </thead>
9   <tfoot>
10    <tr>
11      <td><i>images</i></td>
12      <td><i>régions</i></td>
13      <td><i>départements</i></td>
14    </tr>
15  </tfoot>
16  <tbody>
17    <tr>
18      <td><p><img src=
19        "sourisDansante.gif" width="68" height
20        "83" border="4" alt="petit rat"
21      ></p></td>
22      <td>Ouest</td>
23      <td>Nde</td>
24    </tr>
25    <tr>
26      <td><p></p></td>
29      <td>Centre</td>
30      <td>Nyong et so'o</td>
31    </tr>
32    <tr>
33      <td><p></p></td>
36      <td>Extreme Nord</td>
37      <td>Mokolo</td>
38    </tr>
39  </tbody>
40</table>
```

Le résultat de ce code HTML est donné comme suit :

entete de tableau 1	entete de tableau 2	entete de tableau 3
	Ouest	Nde
	Centre	Nyong et so'o
	Extreme Nord	Mokolo
images	régions	départements

NB : Les balises de fermeture pour tous les éléments de l'entête, du pied ou du corps du tableau sont facultatives mais vivement recommandées.

3. Mise en forme d'un tableau

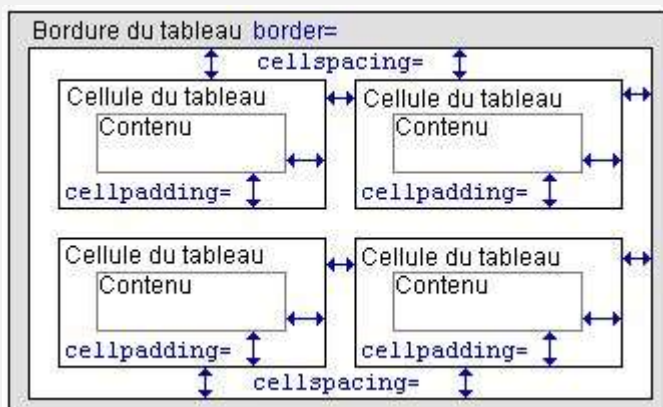
La mise en forme d'un tableau consiste à modifier l'apparence dudit tableau. On peut le faire suivant plusieurs axes :

- ⊕ La définition des espaces entre les cellules , entre le contenu et le bord de la cellule
- ⊕ La définition des bordures extérieures
- ⊕ La définition du quadrillage
- ⊕ La définition des largeurs et hauteurs des colonnes
- ⊕ La définition des passages à la ligne dans les cellules
- ⊕ Fusion et alignement des cellules
- ⊕ La définition des couleurs et images d'arrière-plan
- ⊕ Titre d'un tableau
- ⊕ La mise en forme des tableaux avec CSS

Nous n'allons présenter que quelques unes de ces configurations.

a. Espaces entre les cellules et entre le contenu et le bord de la cellule

Pour mieux comprendre les attributs de la balise `<table>` et leurs effets, observer l'image ci-dessous



L'attribut `cellpadding` permet de fixer l'espace du contenu de la cellule avec le bord de la cellule en pixels (`cellpadding = espace intérieur de la cellule`).

L'attribut `cellspacing` permet de fixer l'espace des cellules entre elles en pixels (`cellspacing = espace entre cellules`).



Quand on crée une bordure avec `border` et un quadrillage visible dans le tableau, l'effet de `cellpadding` et de `cellspacing` peut être mieux retracé. Pour les tableaux sans quadrillage, les deux attributs ne créent rien d'autre que de l'"espace".

b. Bordure extérieure du tableau

La mise en forme de la bordure d'un tableau se fait grâce à l'attribut `border` à l'intérieur de la balise `<table>`.

L'attribut **frame** de la balise **<table>** permet de préciser sur quels côtés du tableau la bordure définie sera appliquée (**frame = cadre**). Les valeurs qu'elle peut prendre sont les suivantes :

- ⊗ **frame="box"** (**box = rectangle**) : la bordure du tableau est visible des quatre cotés (en haut, à gauche, à droite et en bas).
- ⊗ **frame="void"** (**void = rien**) : aucune bordure de tableau n'est affichée. Si on mentionne **border**, le quadrillage des cellules est cependant affiché de façon visible.
- ⊗ **frame="above"** (**above = au dessus**) : une bordure n'est affichée que sur le côté supérieur du tableau.
- ⊗ **frame="below"** (**below = en dessous**) : une bordure n'est affichée que sur le côté inférieur du tableau.
- ⊗ **frame="hsides"** (**hsides = horizontal sides = côtés horizontaux**) : une bordure n'est affichée que sur les côtés inférieur et supérieur du tableau.
- ⊗ **frame="vsides"** (**vsides = vertical sides = côtés verticaux**) : une bordure n'est affichée que sur les côtés droit et gauche du tableau.
- ⊗ **frame="lhs"** (**lhs = left hand side = à gauche**) : une bordure n'est affichée que sur le côté gauche du tableau.
- ⊗ **frame="rhs"** (**rhs = right hand side = droite**) : une bordure n'est affichée que sur le côté droit du tableau.

c. Largeurs et hauteurs des colonnes

Le paramétrage de la largeur se fait avec l'attribut **width**; **height** permet de paramétrer la hauteur.

Étant donné que l'attribut **width** dans une cellule du tableau s'applique à toutes les cellules de la colonne, on n'a besoin de l'utiliser qu'une fois par colonne. Le mieux est de noter cette mention dans la première ligne du tableau. Il en va de même avec l'attribut **height** dans une cellule de tableau.

Étant donné que cet attribut s'applique à toute la rangée de tableau concernée, le mieux est de la noter dans la première cellule de la rangée concernée.

d. Couleurs d'arrière plan

Grâce à l'attribut **bgcolor** dans la balise **<table>**, elle permet de déterminer une couleur d'arrière plan pour le tableau complet (**bgcolor = background color = couleur d'arrière plan**). De la même façon on peut utiliser l'attribut **bgcolor** dans la balise **<tr>** d'une ligne de tableau pour fixer une couleur d'arrière-plan à toutes les cellules de cette ligne. Si on note **bgcolor** dans la balise **<td>** d'une cellule de données ou dans la balise d'une cellule d'entête **<th>** la couleur d'arrière plan s'applique pour cette seule cellule. En cas de conflit la mention de couleur dans une cellule distincte a priorité sur la mention de couleur pour la ligne complète, ou pour tout le tableau.

e. Fusion des cellules

Avec l'attribut `colspan= "nombre de colonnes"`, on obtient qu'une cellule s'étende sur plusieurs colonnes (*colspan = column span = tendre des colonnes*). La mention n'est efficace que si le tableau comprend au moins autant de colonnes qu'il est mentionné.

Avec l'attribut `rowspan= "nombre de rangées"` on obtient qu'une cellule dans une colonne s'étende sur plusieurs rangées (*rowspan = tendre des rangées*). La mention n'est efficace que si le tableau comprend au moins autant de rangées qu'il est mentionné.

Il est possible de combiner les attributs `colspan= "nombre de colonnes "` et `rowspan= "nombre de rangées"` dans une cellule. Cela vous permet d'étendre la cellule sur plusieurs rangées et sur plusieurs colonnes.

Application : le code html ci-dessous permet d'appliquer les points b), c), d) et e)

```
<body>
<table border="3" cellspacing="10"
cellpadding="20" frame="hsides" style=
"background-color:#FFFF00">
<tr>
<th colspan="2">L'humanité est
composée de</th>
<td>femmes</td>
<td>hommes</td>
</tr>
<tr>
<th rowspan="2">Le monde des singes
comprend</th>
<td>les vrais singes</td>
<td colspan="2">Très sensibles</td>
</tr>
<tr>
<td>les humains</td>
<td colspan="2">Vrai ou faux</td>
</tr>
<tr>
<td rowspan="2">Le monde des singes comprend</td>
<td>les vrais singes</td>
<td colspan="2">Très sensibles</td>
</tr>
<tr>
<td>les humains</td>
<td colspan="2">Vrai ou faux</td>
</tr>
</table>
</body>
```

Le résultat obtenu est le suivant :

L'humanité est composée de		femmes	hommes
Le monde des singes comprend	les vrais singes	Très sensibles	
	les humains	Vrai ou faux	Vrai ou faux



Compléments

Empêcher le passage à la ligne dans les cellules

L'attribut **nowrap** (sans autre affectation de valeur) dans la balise **<th>** empêche un passage à la ligne automatique dans le contenu de la cellule. La cellule et ainsi que la colonne dont elle fait partie sera aussi large qu'il est nécessaire.

Alignement dans les cellules

L'attribut **align** placé dans les balises ouvrantes de définition des cellules (**<th>** et **<td>**) permet spécifier l'alignement du contenu des cellules correspondantes. Cet attribut peut prendre plusieurs valeurs : **align="right"** pour l'alignement à droite, **align="left"** pour l'alignement à gauche et **align="center"** pour centrer le contenu (*align = alignement, left = gauche, center = centré, right = à droite*). Par défaut, le contenu d'une cellule est aligné à gauche. Cet attribut peut également s'appliquer aux balises ouvrantes **<tr>**, **<thead>**, **<tbody>** et **<tfoot>**. Alors c'est toutes les cellules de la partie correspondante du tableau qui seront alignées suivant les valeurs de l'attribut **align**.

Avec **valign="top"** dans la balise d'ouverture, il est possible d'aligner une cellule en haut à gauche ou en bas à gauche avec l'attribut **valign="bottom"**. De même l'attribut **valign="middle"** permet l'alignement au milieu de la cellule (*valign = vertical align = alignement vertical, top = haut, bottom = bas, middle = au milieu*). Cet attribut peut également s'appliquer aux balises ouvrantes **<tr>**, **<thead>**, **<tbody>** et **<tfoot>**. Alors c'est toutes les cellules de la partie correspondante du tableau qui seront alignées suivant les valeurs de l'attribut **align**.

II. LES FRAMES

Les **cadres** sont des outils qui permettent d'obtenir plusieurs fenêtres dans une page web. Il est possible à l'aide de cadres de diviser l'affichage du navigateur en différentes parties définissables librement. Chaque partie peut avoir son propre contenu. Les différentes parties d'affichage (appelées **cadres ou frames**) peuvent avoir un contenu statique ou un contenu changeant (dynamique). Il est possible de créer des liens dans un cadre permettant d'appeler des fichiers qui sont ensuite affichés dans un autre cadre.

Les cadres ne sont pas un élément supplémentaire destiné à effectuer un travail spécifique de traitement de texte mais un élément qui utilise rationnellement les propriétés spécifiques de l'affichage. Les cadres ouvrent des possibilités tout à fait nouvelles de préparer l'information de façon hypertextuelle (c'est à dire de façon non-linéaire).

1. Définition des cadres et jeu de cadres

La définition des cadres se fait grâce à la balise **<frameset>...</frameset>**. L'attribut **rows** permet de diviser la fenêtre d'affichage en rangées (*rows = rangées*). L'attribut **cols** permet de diviser la fenêtre d'affichage en colonnes (*cols = columns = colonnes*). Il faut notamment définir comment le partage de la fenêtre doit se faire.

La commande `<frameset rows="20%,80%">.....</frameset>` permet de faire un partage en deux rangées grâce à l'attribut `rows="20%,80%"`; la rangée supérieure occupera 20% de l'écran d'affichage et la rangée du bas 80%.

De même la commande `<frameset cols="200,*">.....</frameset>` permet de diviser la fenêtre en deux colonnes, de façon à ce que la colonne de gauche ait 200 pixels de large tandis que la droite prendra le reste.

La balise `<frame>.....</frame>` permet de définir dans un jeu de cadres le contenu du (des) cadre(s) concerné(s). L'attribut `src` (*source*) permet de faire l'affectation d'une valeur à cet attribut, vous pouvez mentionner le fichier ou la source qui doit être affiché ensuite dans le cadre correspondant.

La balise `<noframes>...</noframes>` s'exprime lorsque le navigateur n'a pas pu afficher les cadres définis.

2. Propriétés des cadres

Les cadres définis dans une fenêtre peuvent être configurés en modifiant les propriétés telles que la couleur de bordure du cadre, l'épaisseur de la bordure, la taille non modifiable de la fenêtre, l'espace minimal entre le bord de la fenêtre et son contenu, forcer ou empêcher l'affichage des barres de défilement. Nous n'allons pas tous les présenter.

L'attribut `scrolling="yes"` dans la balise `<frame>` permet de forcer la fenêtre d'affichage à être pourvue dans tous les cas de barres de défilement. `scrolling="no"` empêche l'affichage des barres de défilement. De la même façon il est possible de mentionner `scrolling="auto"`, quoique ce soit identique au réglage par défaut.



Si vous empêchez qu'une fenêtre soit pourvue de barres de défilement, il est possible que les contenus, s'ils sont plus grands que la fenêtre, ne soient pas entièrement affichés. C'est pourquoi n'utilisez `scrolling="no"` que si vous êtes sûr que le contenu de la fenêtre tiendra entièrement dans la fenêtre-cadre.

- ⊕ L'attribut `bordercolor` dans la balise `<frameset>` vous permet de déterminer une couleur de bordure pour toutes les bordures de fenêtres. On a par exemple `<frame bordercolor="#FFEEAA">`
- ⊕ L'attribut `noresize` dans la balise `<frame>` empêche l'utilisateur de modifier la taille d'un cadre (`noresize = no resize = pas de modification de la taille`). Cet attribut ne prend aucune valeur
- ⊕ L'attribut `frameborder` permet de gérer la bordure d'un cadre. Il prend les valeurs 1/0 ou yes/no. `Frameborder="1"` permet d'afficher de façon explicite la bordure. `Frameborder="no"` permet de rendre la bordure invisible
- ⊕ L'attribut `framespacing` permet de déterminer l'espace entre les fenêtres-cadres, et par conséquent l'épaisseur de la bordure en pixels (`spacing = espace`). `Framespacing = "valeur pixel"`.

3. Liens à des cadres

Cette section présente comment organiser la navigation dans une fenêtre préalablement sectionnée en cadre. Il suffit simplement d'appliquer la partie du cours qui porte sur les liens hypertexte. Pour aller droit au but, voici un bout de code [test.html](#) qui

crée deux cadres dans une fenêtre ; dans la fenêtre de gauche affiche le contenu de la page html nommée [liens.html](#) ; dans la fenêtre de droite, la page nommée [pagedaccueil.html](#). Les codes de ces deux pages vous sont donnés à la suite.

```
<html>
<head>
<title>Texte du titre universel</title>
</head>
<frameset cols="250,*">
  <frame src="liens.htm" name="Navigation">
  <frame src="pagedaccueil.htm" name="donnees">
  <noframes>
    Votre navigateur ne peut malheureusement pas
  </noframes>
</frameset>
</html>
```

Code du fichier test.html

```
<html>
<head>
<title>Texte du titre</title>
</head>
<body bgcolor="#F0C0FF" text="#000000" link="#804080" vlink="#603060" alink="#804080">
  <h1>Navigation</h1>
  <p>
    <a href="../index.htm" target="donnees"><b>Sur les cadres</b></a><br>
    <a href="../../index.htm" target="donnees"><b>Sur HTML</b></a><br>
    <a href="../../../index.htm" target="donnees"><b>Encore bien plus</b></a>
  </p>
</body>
</html>
```

Code du fichier lien.html

```
<html>
<head>
<title>Texte du titre</title>
</head>
<body bgcolor="#FFF0C0" text="#000000" link="#A00000" vlink="#600000" alink="#A00000">
  <h1>Bienvenue</h1>
  <p>Choisissez un lien à gauche!</p>
</body>
</html>
```

Code du fichier pagedaccueil.html

Le résultat est le suivant :



Travail à faire : tester ces codes puis interpréter ces résultats. Qu'est ce que cela vous inspire ?

REMARQUE

1. La définition d'un frame dans un document html ne nécessite pas l'utilisation de la balise `<body> </body>`. Il faut remarquer que la somme des pourcentages doit être égale à 100%
2. Lorsque qu'on définit un ensemble de cadres avec la balise `</frameset>`, il faut que le nombre de fenêtre (nombre de paramètre de l'attribut `cols` ou `rows` de `frameset`) définie doit correspondre au nombre de balise `<frame>`.

On a par exemple

```
<html>
  <head> <title> Exemple de definition des frame</title> </head>
  <frameset cols="20%,30%,50%">
    <frame> <!--représente les 20% de la fenêtre -->
    <frame> <!-- représente les 30% de la fenêtre -->
    <frame> <!-- représente les 50% de la fenêtre -->
  </frameset>
</html>
```



Résumé

Les **balises** permettent de créer des objets ou d'ajuster des mises en forme. A ces balises sont associés des **attributs**. Les attributs permettent d'apporter un plus à la balise.

Les tableaux sont des objets créés à l'aide des balises `<table>` et `</table>`. Les cellules d'un tableau sont créées par les balises `<th>`, `</th>`, `<td>`, `</td>`, `<tr>`, `</tr>`.

Créer un tableau se résume à configurer les espaces entre les cellules et entre le contenu et le bord de la cellule, les bordures, le quadrillage, les largeurs et hauteurs d'un tableau, les couleurs, les arrières plan, le titre, les entêtes, etc.

Les **cadres** sont des outils qui permettent de d'obtenir plusieurs fenêtres dans une page web. Pour définir les cadres, il faut définir un jeu de cadres à l'aide des balises `<frameset >.....</frame>`, ensuite, définir dans un jeu de cadres le contenu du (des) cadre(s) concerné(s) en utilisant `<frame>.....</frame>`,

La configuration des cadres peut se faire grâce aux balises de modification des bordures (épaisseur, couleur), de l'ajout ou non des barres de défilement.



Exercice 1 :

Quelle est la balise qui permet de créer une table ? Lister et donner le rôle de chacun de ses attributs.

Quelle est la balise qui permet de créer un cadre ? Lister et donner le rôle de chacun de ses attributs.

Exercice 2 :

Écrire un code html qui présente un menu. C'est-à-dire, vous allez insérer un tableau d'une colonne et cinq lignes avec ou sans bordure selon votre convenance ; dans chaque cellule, insérer un texte et le centré.

Pour fournir d'avantage votre code, mettez de l'espace entre les cellules

Exercice 3 :

Écrire un code html dans lequel vous ne saisissez qu'un paragraphe qui décrit votre lycée. Insérer ce paragraphe dans un tableau d'une cellule ayant une bordure de largeur ou épaisseur 7 et de couleur verte.

Exercice 4 :

Écrire un code html qui contient un tableau d'une cellule sans bordure. Dans la cellule insérer une image qui s'ajuste au contenu.

Exercice 5 :

Écrire un code html qui respecte les étapes suivantes :

- ④ Créer deux cadres en vertical tels que le cadre de gauche A recouvre les 20% de la fenêtre et 80% pour le reste (B)
- ④ Dans le cadre de gauche A faire référence au fichier html écrit à l'exercice 2
- ④ Dans le cadre de droite B, créer deux autres cadres dans le sens horizontal B1 et B2 tels que celui du haut (B1) représente les 90% et le cadre du bas (B2) les 10%
- ④ Dans le cadre B1 faire une référence au fichier html écrit à l'exercice 3
- ④ Dans le cadre B2 faire une référence au fichier html écrit à l'exercice 4

Chapitre 12

INSERTION DES FORMULAIRES ET DES BOUTONS DANS UNE PAGE WEB

Nom:
Tel:
Adresse

Vos matières de bases

☒ <i>Maths</i>	☒ <i>Histoire</i>	☐ <i>Physique</i>
☐ <i>Anglais</i>	☐ <i>SVT</i>	☐ <i>Chimie</i>

Votre mention au BEPC:

☒ *Passable*
☐ *assez bien*
☐ *bien*
☐ *T.bien*

SOMMAIRE



- I. INSERTION DES FORMULAIRES
- II. INSERTION DES BOUTONS



Problème

Il vient d'être créé dans votre lycée un site web faisant la présentation de tout son personnel administratif, même sa description physique. Ce site permettra aux élèves et tous autres visiteurs de faire des suggestions, d'avoir des discussions. On aimerait également que pour faire une tierce remarque ou suggestion, que le visiteur donne un certains nombres d'informations sur lui afin de s'enregistrer.

Que faut-il faire pour orienter le visiteur lors de son enregistrement afin qu'il ne donne que l'essentiel



Pré-requis

- Utiliser un éditeur de texte et un navigateur
- Connaître la structure de base d'un document HTML
- Connaître un certains nombre de balise et leur rôle



Objectifs

- Insérer un formulaire simple de données
- Insérer un formulaire de données dans une page web
- Insérer les boutons

INTRODUCTION

Lors de la création de son site web, on peut vouloir créer une rubrique « forum » à travers lequel les différents visiteurs partageront les informations. Pour cela il faut créer des formulaires permettant à ces visiteurs de s'enregistrer afin de pouvoir participer aux différentes discussions.

Lors du remplissage du formulaire ou à la fin du remplissage, le visiteur peut vouloir réinitialiser dans la mesure où il a fait des erreurs, validé pour que le site prenne en compte les informations saisies. Pour ce fait il aura besoin des boutons.

Dans ce chapitre nous allons dans un premier temps insérer les formulaires de données et dans un second temps insérer les boutons.

I. INSÉRER UN FORMULAIRE

Des formulaires peuvent être préparés afin de saisir des données et les traiter au niveau du serveur. Avec les formulaires, le langage Html vous ouvre la porte de l'interactivité. Votre interlocuteur pourra remplir un champ de saisie de texte ou d'autres objets interactifs comme des cases à cocher. Ces données seront transmises au serveur qui peut être votre adresse électronique (Mail).

Pour créer un formulaire on a besoin d'un ensemble de commande tel que :

1. La Balise FORM

C'est la balise qui permet d'insérer un formulaire ou un questionnaire dans une page web. Pour rédiger un questionnaire, il faut:

- ④ Établir une zone d'insertion du formulaire (appelée FORM) en utilisant la balise `<FORM>...</FORM>`.
- ④ Définir la méthode à employer pour transmettre au serveur l'information recueillie dans les champs du formulaire à travers l'attribut `method` de la balise `<FORM>...</FORM>`.
- ④ Identifier l'emplacement et le nom du programme qui devra traiter l'information recueillie à travers l'attribut `action` de la balise `<FORM>...</FORM>`.
- ④ Fournir, s'il y a lieu, les arguments au programme de traitement des données.

2. La balise INPUT

Plusieurs attributs sont définis, parmi lesquels l'attribut `type`, `name`, `id`... l'attribut obligatoire et le plus important dans cette partie est l'attribut `type`. Cet attribut à plusieurs valeurs définissant le type de champ du questionnaire utilisé. Ces valeurs peuvent être `text`, `checkbox`, `radio`, `textarea`.

a. cas du type « TEXT »

Parmi les choix disponibles en HTML, un des types d'entrée de données est le champ `input type="text"`. Dans ce cas, il faut inscrire le type de champ (`text`), le nom du champ (par l'attribut `name`) et ses dimensions à l'écran (par l'attribut `size`).

Exemple l'instruction ci-dessous permet au visiteur d'entrer son nom:

NOM: `<input type="text" name="name" size=30>`.

Le type `text` permet de créer un champ où l'utilisateur entre l'information requise. L'attribut `size` permet de définir la longueur de la zone de saisie qui ici est d'une longueur de `30` espaces. Pour un champ plus long, entrez une valeur plus grande. La commande `name="name"` précise que l'on désire enregistrer le contenu du champ dans la rubrique `"name"`.



Création des « TEXT » dans un formulaire

Etape1 : ouvrir un éditeur de texte (Notepad++, Bloc-notes, ...) et saisir le code suivant

```
<HTML>
<FORM METHOD="POST" ACTION="taritament.php">
<p><h2> Essai Formulaire </H2><pre><b>
<LI>NOM: <input type="text" name="name" size=30><br/>
<LI>ADRESSE: <input type="text" name="street.address" size=30>
<br/><LI>PROVINCE: <input type="text" size=15 name="state">
<br/><LI>PAYS:<input type="text" name="country" size=20>
<br/></b></pre></form><p>
</BODY>
</HTML>
```

Etape2 : enregistrez votre travail sous l'extension .html et visualisez à l'aide d'un navigateur

b. Cas du type « RADIO »

Un autre type de champ est le type `"input type=radio"` qui permet d'afficher une série de boutons comme choix de réponses. Dans ce type de bouton, on ne peut faire qu'un seul choix à la fois. Il suffit d'abord de poser la question puis de positionner la commande.

Exemple : soit à insérer cette question dans un formulaire :

`<p>êtes-vous majeur ou mineur ? </p>`

`<input type="radio" name="info" value="Majeur">Majeur (18 ans et plus)`

`<input type="radio" name="info" value="Mineur">Mineur (moins de 18 ans)`

L'attribut `« value »` permet de spécifier une possibilité de réponse. Dans l'activité ci-dessous, deux possibilités de réponse sont offertes **Majeur** ou **Mineur**. Lorsque l'utilisateur répond, il ne peut opérer qu'un seul choix.

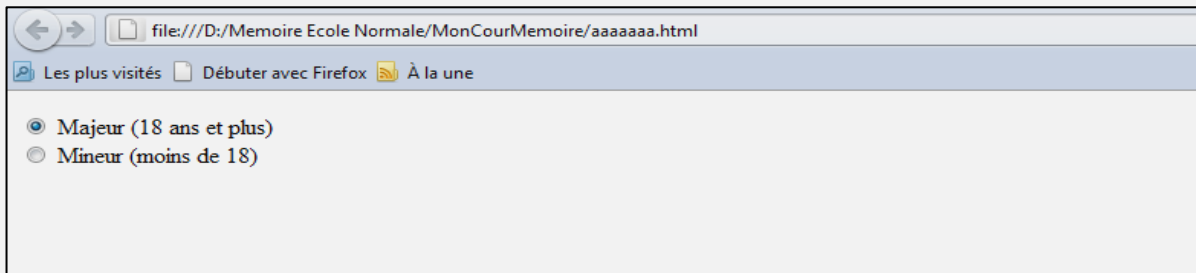


Création des « radios » dans un formulaire

Etape1 : ouvrir un éditeur de texte (Notepad++, Bloc-notes, ...) et saisir le code suivant

```
<HTML>
<FORM METHOD="POST" ACTION="traitement.php"> <p>
<input type="radio" name="Age" value="majeur" id="majeur">
<label for="majeur"> Majeur (18 ans et plus) </label><br/>
<input type="radio" name="Age" value="mineur" id="mineur">
<label for="mineur"> Mineur (moins de 18) </label> <br/> </p>
</form>
</BODY>
</HTML>
```

Etape2 : enregistrez votre travail sous l'extension .html et visualisez à l'aide d'un navigateur



c. Cas du type « CHECKBOX »

La commande input **Type=checkbox** permet d'afficher une liste où plusieurs choix sont possibles en même temps. Selon qu'on souhaite afficher le bouton avant ou après le texte, la commande s'écrit comme suit:

Texte `<input name="nom_du_champ" TYPE=checkbox VALUE="texte">`
OU

`<input name="nom_du_champ" TYPE=checkbox VALUE="texte">` *Texte*

Exemple :

Ⓢ Si on veut afficher le texte avant, on a :

BTS `<input name="Formation" TYPE=checkbox VALUE="BTS">`

Ⓢ Si on veut afficher le texte après, on a :

`<input name="Formation" TYPE=checkbox VALUE="BTS">` **BTS**



Création des « checkbox » dans un formulaire

Etape1 : ouvrir un éditeur de texte (Notepad++ , Bloc-notes, ...) et saisir le code suivant

```

<html>
<body><form method ="post" action ="traitement.php">
<p><h1>Quels marchés avez vous déjà visités </h1> <br/>
<input type ="checkbox" name="Marché de Mokolo" id="Marché de Mokolo">
<label for="Marché de Mokolo"> le Marché de Mokolo </label> <br/>
<input type="checkbox" name="Marché Central" id="Marché Central">
<label for="Marché Central"> le Marché Central </label> <br/>
<input type="checkbox" name="Marché de Mendong" id="Marché de Mendong">
<label for="Marché de Mendong"> le Marché de Mendong </label> <br/>
</p></form></body>
</html>

```

Étape 2 : enregistrez puis visualisez à l'aide d'un navigateur

Quels marchés avez vous déjà visités

- ☒ le Marché de Mokolo
- ☐ le Marché Central
- ☒ le Marché de Mendong

3. Les balises SELECT NAME et OPTION

La commande *select name* et *Option* permettent d'afficher une liste déroulante où un seul choix est possible. La commande s'écrit:

`<select name> <option selected>choix1<option>choix2... <option>choixn </select>`

Dans cette syntaxe `<select name>` est la balise ouvrante de la liste déroulante. `<option>` permet de citer les choix possibles et l'attribut *selected* permet de définir la valeur qui s'affichera par défaut (c'est le choix1 dans la syntaxe ci-dessus).

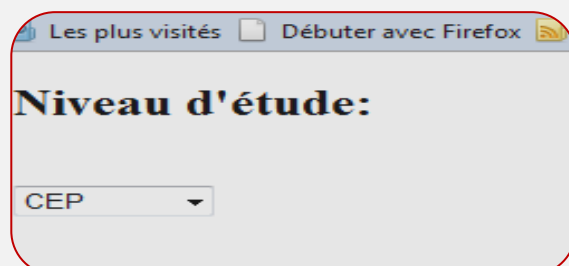


Création des listes déroulantes dans un formulaire

Étape 1 : ouvrir un éditeur de texte et saisir le code suivant

```
<HTML>
<FORM METHOD="POST" ACTION="traitement.php"> <p>
<h2> <b>Niveau d'étude: </b></h2><BR>
<select name><option selected>CEP<option>BEPC<option>Probatoire<option>BAC</select>
</form>
</BODY>
</HTML>
```

Étape 2 : enregistrez et visualisez :
Vous obtiendrez un résultat semblable à
Celui-ci.



4. La balise TEXTAREA

Cette commande permet de créer une zone de saisie avec des dimensions de votre choix. La syntaxe utilisée est la suivante:

`<TEXTAREA name="nom_du_champ" rows=n cols=m></TEXTAREA>`

Dans ce type de commande, on spécifie d'abord le type soit *textarea*, ensuite l'attribut `name="nom_du_champ"` précise que l'on désire enregistrer le contenu du champ dans la rubrique "*nom_du_champ*". On précise enfin les dimensions de la zone de saisie à l'aide des attributs `rows` pour le nombre de lignes (n) et `cols` pour le nombre de colonnes (m).

Exemple : `<TEXTAREA name="zone_libre" rows=5 cols=40></TEXTAREA>`

Après exécution de cet exemple, on aura une zone de texte de 5 lignes et de 40 colonnes.



Insertion d'une zone de saisie dans une page Web

Étape 1 : ouvrir un éditeur de texte et saisir le code suivant

```
<HTML>
<BODY >
<FORM METHOD="POST" ACTION="traitement.php">
  <h2> veuillez raconter une histoire</h2>
  <label for="histoire" ></label><br/>
  <textarea name="histoire" id="histoire" rows="5" cols="40"> </textarea>
</form> </BODY> </HTML>
```

Étape 2 : enregistré sous l'extension *.html* et visualiser. Vous allez obtenir un résultat semblable à celui ci-contre.

veuillez raconter une histoire

il était une fois, une princesse ...

Les formulaires doivent contenir des boutons permettant soit d'envoyer le contenu des champs remplis au serveur http, soit de réinitialiser les champs des formulaires.

II. INSERTION DES BOUTONS

Le bouton est un élément d'une page web ou d'un formulaire permettant de réaliser une action bien définie. Dans un formulaire, il peut être utilisé soit pour remettre celui-ci à l'état initial, soit pour valider une opération effectuée, soit pour accéder à une autre page etc....

Nous allons dans cette partie présenter quelques boutons.

I. Les boutons d'envoi

Ce bouton commande l'envoi des données. Après avoir cliqué dessus le visiteur est renvoyé vers une autre page prédéfinie par le concepteur du site. La balise permettant d'insérer le bouton est la balise `<input>` avec un attribut obligatoire `type` qui prend la valeur « `submit` ». cette valeur permet de spécifier qu'il s'agit d'un bouton d'envoi. La syntaxe est la suivante : `<input type="submit">`.

Le texte qui s'affiche par défaut sur le bouton est « `envoyer` ». Si l'on désire modifier ce texte, il suffit d'ajouter l'attribut `value` dans la balise `<input>` avec comme valeur l'expression qu'on veut afficher sur le bouton.

La syntaxe est la suivante : `<input type="submit" value="expression">`.



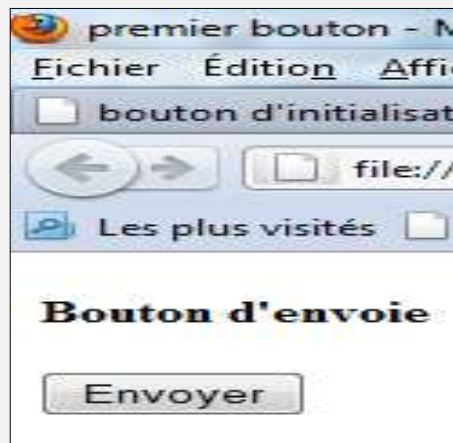
Insertion d'un bouton d'envoi dans une page Web

Étape 1 : ouvrir un éditeur de texte et saisir le code ci -contre

```
<html>
<head><title> premier bouton </title></head>
</body>
<form method="post" action="traitement.php">
<h4>Bouton d'envoi </h4>
<input type="submit">
</form>
</BODY>
</HTML>
```

Étape 2 :

Enregistrer le travail sous l'extension `.html` et visualisez votre résultat. Il doit être semblable à la figure ci-après



2. Le bouton de mise a zéro ou de réinitialisation

Comme son nom l'indique, ce bouton remet tous les éléments du formulaire à leur valeur par défaut. Il peut arriver par exemple à un moment donné que l'on coche un ensemble d'éléments d'un formulaire et qu'on veuille tout changer, ce bouton permettra de tout réinitialiser en un clic. La syntaxe est la suivante : `<input type="reset">`. Le texte qui s'affiche par défaut sur le bouton est « effacer ». Si l'on désire modifier ce texte, il suffit d'ajouter l'attribut `value` dans la balise `<input>` avec comme valeur l'expression qu'on veut afficher sur le bouton.

La syntaxe est la suivante : `<input type="reset" value = " expression ">`.



Insertion du bouton Effacer dans une page Web

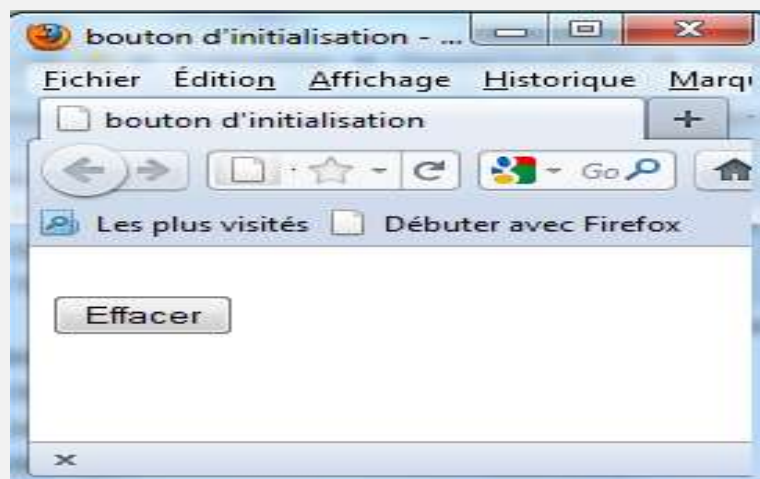
Étape 1 : ouvrir un éditeur de texte (Notepad++, Bloc-notes, ...) et saisir le code suivant

```

1  <html>
2  <head>
3  <title> bouton d'initialisation </title>
4  </head>
5  </body>
6  <br/>
7  <form method="post" action="traitement.php">
8  <input type="reset">
9  </form>
10 </body>
11 </html>

```

Étape 2 : enregistrer votre code sous l'extension .html et le visualiser avec un navigateur. Vous obtiendrez une image semblable à celle ci-dessous





Compléments

On peut également utiliser une image comme un bouton. Pour cela il suffit d'affecter la valeur « image » à l'attribut **type**, et remplacer l'attribut **value** par l'attribut **src** qui spécifie le chemin d'accès à l'image. On peut aussi définir la taille que l'image prendra sur notre page web. Pour cela on utilise les attributs **"width"** et **"height"**.

Le code est le suivant:

```
<html>
<head>
  <title>Mon titre</title>
</head>
<body>
  <h1>Ma Photo</h1>
  <form Method="post" action="http://www.ens.cm">
    <input type="image" src="nom_Image.gif" width="50" height="10">
  </form> </body> </html>
```



Pour gagner en temps

Lorsque vous êtes amené à saisir chaque fois le code HTML, il est judicieux de garder dans un fichier sa structure. Ainsi chaque fois que vous auriez besoin de saisir du code HTML, vous feriez un copie-collé afin de gagner en temps de saisis.



Résumé

Un formulaire est très important dans un site. Pour l'insérer dans une page web, on utilise la balise **<form> ... </form>**. Dans cette balise on utilise d'autres balises permettant de créer des zones de texte, des listes déroulantes, des questions à choix multiple ou à choix unique...

- ② la balise **< textarea >...</textarea >** permet de créer une zone de texte sur plusieurs colonnes et lignes.
- ② La balise **input** qui permet d'inclure un certain nombre de type tel que le **type="text"** pour une zone de saisir d'une seule ligne ; le **type = "radio"** pour des questions à choix unique ; le **type = "checkbox"** pour des questions à plusieurs choix, le type **submit** et **reset** qui permettent respectivement d'insérer un bouton d'envoi et un bouton de réinitialisation.
- ② La balise **<select name>...</select> et <option>** pour l'insertion des **listes déroulantes** permettant d'effectuer un seul choix parmi plusieurs.



Exercice 1

Créer un formulaire qui permettra de faire la collecte des informations sur les élèves du Lycée de Garoua au Nord. Comme information, il faut le nom, le prénom, date de naissance, lieu de naissance, classe et nom du professeur titulaire.

Exercice 2

Créer un questionnaire, chacune des questions suivies de plusieurs réponses au choix donnant la possibilité à l'interlocuteur ou emmenant l'interlocuteur de faire son ou ses choix. Les questions sont les suivantes :

- a) Lycée actuellement fréquenté.
- b) Professeur titulaire
- c) Êtes-vous mineur ou majeur ?
- d) Quelles villes avez-vous déjà visités ?
- e) Dans quel(s) marché(s) avez-vous déjà faire des achats ?
- f) Quelles sont vos matières préférées ?
- g) Votre nationalité (NB : a cette question il faudra proposer au moins 10 pays).

Exercice 3

Créer un autre formulaire qui donne à l'utilisateur une zone de saisir lui permettant de relater une histoire ou de faire des suggestions. Cette zone doit avoir comme nombre de colonne 50 et comme nombre de ligne 60.

Exercice 4

Reprendre l'exercice 2 et y ajouter un bouton qui permettra à l'utilisateur de tout réinitialiser si le besoin se fait sentir. Sur ce bouton doit être écrit « *remise à zéro* ».

Exercice 5

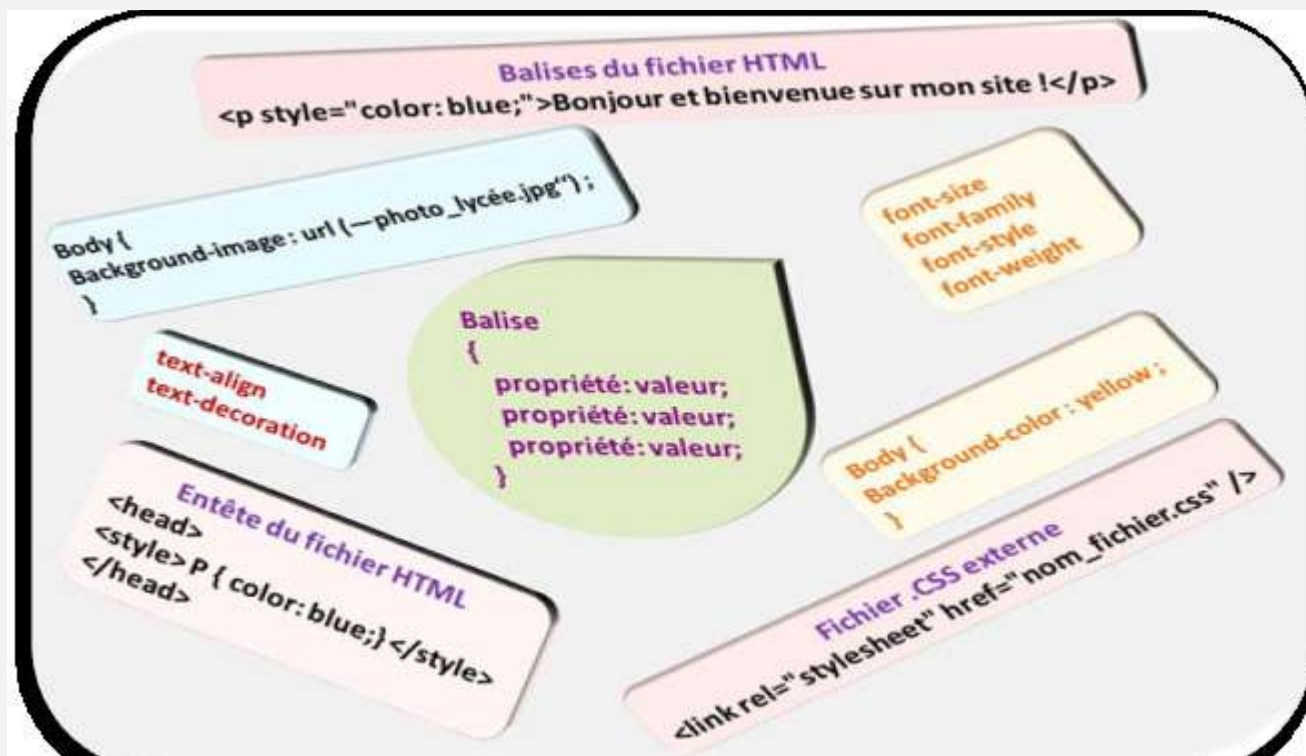
Reprendre l'exercice 4 et y ajouter un bouton qui permettra à l'utilisateur d'envoyer ces informations au serveur. Sur ce bouton doit être écrit « *valider et soumettre* ».

Exercice 6

Combiner les exercices 1, 3 et 5 pour avoir un formulaire plus complet.

Chapitre 13

LES FEUILLES DE STYLE



SOMMAIRE



- I. COMMENT APPLIQUER UNE FEUILLE DE STYLE ?
- II. QUELQUES PROPRIÉTÉS DE FORMATAGE DU TEXTE
- III. MODIFICATION DE L'ARRIÈRE PLAN DE LA PAGE
- IV. OÙ PLACER LE CODE "CSS" ?



Problème

Vous faites partie du groupe chargé de la réalisation d'un site web pour votre lycée. Après avoir créé quelques pages pour votre site, vous constatez qu'elles ne sont qu'en noir et blanc. Vous souhaitez donc les agrémenter non seulement en ajoutant des couleurs, mais aussi avec des images, couleurs d'arrière plan, des bordures etc. Comment procéderez-vous pour la mise en forme de votre site web ?



Pré-requis

- Connaître la structure de base d'un document HTML ;
- Savoir utiliser les balises et leurs attributs ;
- Créer et enregistrer un document dans un éditeur de texte avec une extension précise ;
- Connaître quelques balises de structuration du texte ;



Objectifs

- Définir et donner le rôle du CSS ;
- Savoir les différents lieux de positionnement du code CSS ;
- Créer un fichier de style externe au document HTML ;
- Ajouter l'attribut style à une balise
- Insérer une balise "style" dans l'entête du document ;
- Changer le background d'une page HTML.

INTRODUCTION

Les chapitres précédents portaient sur la création des pages web d'un site, l'organisation des contenus, l'insertion de liens, de tableaux, des cadres etc. Bien qu'on ait présenté quelques balises et attributs permettant d'ajouter des couleurs, celles-ci ne sont pas suffisantes pour donner ces aspects attractif et dynamique qu'on connaît aux sites web. Pour cela, il faut utiliser **les feuilles de styles** encore appelées **CSS (Cascading Style Sheets)**. C'est un langage qui permet de réaliser la **mise en page** et le **graphisme** de la page web. Si **HTML** s'occupe de créer le fond ou le contenu, **CSS** s'occupe de la mise en forme et donne au contenu l'apparence qu'on souhaite. Il permet de choisir la couleur, la taille, le type de la police du texte, d'appliquer des bordures, des fonds etc.

I. COMMENT APPLIQUER UNE FEUILLE DE STYLE ?

La structure générale d'un code CSS est la suivante :

Élément : peut être une **balise**, **plusieurs balises**, ou alors un **sélecteur**. C'est les noms des balises dont on veut modifier l'apparence.

Propriété : ce sont les "effets de style" de la page. Il y a par exemple la propriété **color** qui permet d'indiquer **la couleur du texte**, **font-size** qui permet d'indiquer **la taille du texte**.

Élément

```
{
  propriété: valeur;
  propriété: valeur;
  propriété: valeur;
}
```

Les valeurs : à chaque propriété CSS on doit indiquer une valeur. Par exemple, pour la couleur, il faut indiquer le nom de la couleur. Pour la taille, il faut indiquer quelle taille on veut.

1. Appliquer un style à une balise

Dans ce cas, **element** est une **balise**. On a la syntaxe ci-dessous.

Il convient de noter que dans un fichier CSS, on n'a pas besoin d'écrire le nom de la balise avec les signes '<' et '>'

Exemple : `P { font-size : 16 px }`

Ce code CSS signifie que tous les paragraphes seront écrits avec une taille de 16 pixels.

Balise

```
{
  propriété: valeur;
  propriété: valeur;
  propriété: valeur;
}
```

2. Appliquer un style à plusieurs balises

Dans ce cas, **element** est un ensemble de balises sur lesquelles on souhaite appliquer une même mise en forme.

```
Balise1, balise2,..., balise n
{
  propriété: valeur;
  propriété: valeur;
  propriété: valeur;
}
```

Exemple :

```
P, h2
{
  Color :blue ;
  font-style: italic;
}
```

Ce code signifie que tous les paragraphes, les titres de niveau 2 sont en italique et de couleur bleu.

3. Appliquer un style avec un sélecteur

Ce qui a été fait jusqu'ici présente tout de même un défaut : cela implique que les contenus des balises de même type auront la même mise en forme. Comment faire pour appliquer cette mise en forme seulement aux contenus de certaines balises de même type?

Pour cela, on peut utiliser des **attributs spéciaux** ou des **sélecteurs** qui fonctionnent sur toutes les balises : l'**attribut "class"** et l'**attribut "id"**.

a. Attribut 'class'

C'est un attribut que l'on peut mettre sur n'importe quelle balise, aussi bien **titre** que **paragraphe**, **image**, etc. Cet attribut permet d'écrire un nom (n'importe lequel tant qu'il commence par une lettre) qui sert à identifier la balise sur laquelle sera appliquée la mise en forme. C'est ce nom qui est utilisé dans le fichier CSS en lieu et place de la balise.

Exemple : soit le bout de code HTML suivant :

```
<p class="introduction"> Bonjour
et bienvenue sur mon site </p>
```

Le code CSS permettant de mettre en rouge ce paragraphe :

```
.introduction
{ color: red; }
```

N.B : on place un point devant le nom identifiant la balise

b. Attribut 'id'

Il fonctionne exactement de la même manière que 'class', à la différence qu'il ne peut être utilisé qu'une fois dans le code. En pratique, on ne met des 'id' que sur des éléments qui sont uniques sur la page web, comme le logo par exemple. Quand on utilise l'attribut 'id', il faudra, dans le CSS, faire précéder le nom de l'id par un dièse (#) et non un point (.) comme dans le cas de l'attribut class.

Exemple : Reprenons l'exemple ci-dessus

En utilisant l'attribut 'id' :

`<p id="introduction"> Bonjour et bienvenue sur mon site </p>`

Le code CSS permettant de mettre en rouge ce paragraphe :

```
#introduction
{ color: red; }
```

N.B : l'attribut `id="introduction"` ne peut apparaître qu'une seule fois dans la page.

APPLICATION :

1. Quelle est la différence entre l'attribut class et l'attribut id ?
2. Écrire le code d'un fichier CSS appliquant une couleur précise à un titre et à un paragraphe.



Compléments

L'utilisation de **class** et **id** est limitée du fait qu'ils sont des attributs, et de ce fait ne sont applicables qu'à l'intérieur des balises. Or, il arrive parfois qu'on ait besoin d'appliquer une class (ou un id) à certains mots qui ne sont pas à l'origine entourés par des balises. Par exemple, on peut vouloir appliquer une couleur juste au mot « Bienvenue » dans l'instruction suivante `<p> Bonjour et bienvenue sur mon site </p>`

Pour ce fait, il existe 2 balises dites universelles qui n'ont aucune signification particulière.

Ⓢ `<span> ... </span>` : C'est une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises `<strong>` et `<em>` sont de la même famille.

Ⓢ `<div> ... </div>` : c'est une balise qui entoure un bloc de texte. Les balises de la même famille sont `<p>`, `<hr>`, etc.

Avec l'exemple ci-dessus on utilisera la balise universelle `<span>...</span>` : `<p> Bonjour et <span class="introduction"> Bienvenue </span> sur mon site </p>`

II. QUELQUES PROPRIÉTÉS DE FORMATAGE DE TEXTE :

Propriétés CSS	Rôles	Valeurs possibles	Exemples
font-size	Taille de la police	taille absolue : en pixels, en centimètres ou millimètres taille relative : en pourcentage, "em" ou "ex"	P { Font-size : 1.3 em }
font-family	Type de police	Arial, Arial Black, Comic Sans MS, Courier New, Georgia,	P { Font-family: "Arial"

		Impact, Times New Roman, etc.	Black'' ;}
font-style	Texte en italique	italic : le texte en italique. oblique : le texte en italique (en penchant les lettres). normal : le texte sera normal	H1 { Font-style : oblique ; }
font-weight	Texte en gras	bold : texte en gras normal : texte normal	span { Font-weight : bold ; }
text-decoration	Texte souligné, barré etc.	underline : souligné. line-through : barré. overline : ligne au-dessus. blink : clignotant. none : normal	H3 { Text-decoration : overline ; }
text-align	Alignement du texte	left : alignement à gauche center : alignement au centre right : alignement à droite. justify : le texte sera "justifié"	P{ Text-align : justify ; }
Color	La couleur du texte	Indiquer le nom de la couleur en anglais	P{ color : green ; }

APPLICATION :

1. À quoi servent les balises universelles ? citer les et donner la différence qui existe entre elles ?
2. Écrire le code d'un fichier CSS permettant de mettre en gras, italique et faire clignoter un mot particulier d'un paragraphe

III. MODIFICATION DU FOND DE LA PAGE

On peut soit modifier la couleur, soit mettre une image à l'arrière plan d'une page web.

1. Modification de la Couleur de fond

Pour indiquer une couleur de fond, on utilise la propriété CSS **background-color**. Elle s'utilise de la même manière que la propriété **color**. On peut taper le **nom d'une couleur**, l'écrire en **notation hexadécimale** ou encore utiliser **la méthode RGB**.

Pour indiquer la couleur de fond de la page web, il faut travailler sur la balise **<body>**, car elle correspond à toute la page web, c'est donc en modifiant sa couleur de fond que l'on changera la couleur de fond de la page web.

```
Body
{
Background-color : yellow ;
}
```



On peut changer le fond de n'importe quel élément : titres, paragraphes, certains mots, etc. Dans ce cas, ils apparaîtront surlignés (comme si on avait mis un coup de marqueur dessus). Pour cela, on utilise la balise **<mark>...</mark>** avec la propriété **background-color**.

Exemple: `mark {background-color : red ;}`

2. Insertion d'une image de fond

La propriété permettant d'indiquer une image de fond est **background-image**. L'image doit être dans le dossier du site. Comme valeur, on doit lui donner **url("nom_de_l_image.png")**.

Exemple : `Body { Background-image : url ("photo_lycée.jpg") ; }`

IV. OÙ PLACER LE CODE "CSS" ?

On peut écrire un code en langage CSS à 3 endroits différents :

- ② Dans un **fichier .css externe** au document HTML (c'est la méthode la plus recommandée);
- ② Dans **l'en-tête <head>** du fichier HTML ;
- ② Directement dans **les balises du fichier HTML** via un attribut style (méthode la moins recommandée).

1. Dans un fichier externe .css

Dans ce cas, on écrit le code CSS dans un fichier spécial ayant l'extension **.css** (contrairement aux fichiers HTML qui ont l'extension **.html**). C'est la méthode la plus pratique et la plus flexible, car il permet d'éviter de tout mélanger dans un même fichier. Ayant 02 fichiers, il faut **un lien** qui indique que ce fichier HTML va être associé à un fichier externe (**.css**) qui va gérer sa **mise en forme**. L'instruction permettant d'insérer ce lien est la suivante : **<link rel="stylesheet" href="nom_fichier.css" />**

EXEMPLE : Soit le fichier de mise en forme nommé **style.css** suivant :

```
P {
color: blue;
}
```

L'en-tête du document HTML où il sera appliqué est la suivante:

```
<head>
<link rel="stylesheet" href="style.css" />
<title> ma page </title> </head>
```

Il convient de noter que les 02 fichiers doivent être placés dans un même dossier et le lien ci-dessus est placé dans **l'entête du fichier HTML**.

La balise non fermante qui permet d'insérer le lien est **<link />**

L'**attribut rel** permet de spécifier le **type de fichier** et l'**attribut href** permet de donner le **nom du fichier CSS** correspondant.

2. Dans l'entête du fichier HTML

Cette méthode consiste à insérer le code CSS directement dans une balise **<style>...</style>** à l'intérieur de l'en-tête du fichier HTML c'est-à-dire l'intérieur de la balise **<head>...</head>**.

EXEMPLE : On désire mettre en bleu tous les paragraphes de notre page grâce à une feuille de style :

```
<head>
<style> P { color: blue;} </style>
<title> ma page </title> </head>
```

3. Dans les balises du fichier HTML

Cette dernière méthode stipule qu'on peut ajouter un **attribut style** à n'importe quelle balise. Pour cela, on insère le code CSS directement dans cet attribut. Cette fois, seul le contenu de la balise qui contient le code CSS est mis en forme.

EXEMPLE:

```
<p style="color: blue;"> Bonjour et bienvenue sur mon site ! </p>
```



OÙ PLACER LE CODE CSS ?

Étape 1 : Fichier de mise en forme externe au document HTML

Lancer l'éditeur de texte et créer 02 documents : un document html **ma_page.html** et un fichier de mise en forme de cette page **style.css** avec les codes suivants :

```
Fichier Edition Recherche Affichage Format Langage Paramétrage Mac
ma_page.html style.css
1 <html>
2 <head>
3 <link rel="stylesheet" href="style.css" />
4 <title> premier test CSS </title>
5 </head>
6 <body>
7 <h1>Message de bienvenue </h1>
8 <p>Bonjour et bienvenue sur mon site!</p>
9 <h1> Algorithmique </h1>
10 <p> notion de base en algorithmique </p>
11 <h1> Infographie </h1>
12 <p> les applications de l'infographie </p>
13 <p> création des <em> boutons </em> </p>
14 </body>
15 </html>
```

```
Fichier Edition Recherche Affichage Format Langage
ma_page.html style.css
1 h1
2 {
3 color:blue;
4 }
5 p
6 {
7 font-family: Impact;
8 font-size: 16px;
9 }
```

Placer les 02 documents dans un même dossier et ouvrir la page web **ma_page.html** pour le tester. On peut constater que tous les textes de la balise `<h1>` sont de couleur bleu.



Étape 2 : Fichier de mise en forme à l'entête du document HTML

À partir du code HTML ci-dessus, créer une nouvelle page nommée **ma_page2.html** ; insérer dans son entête un code CSS qui va mettre en rouge tous les paragraphes de la page web.

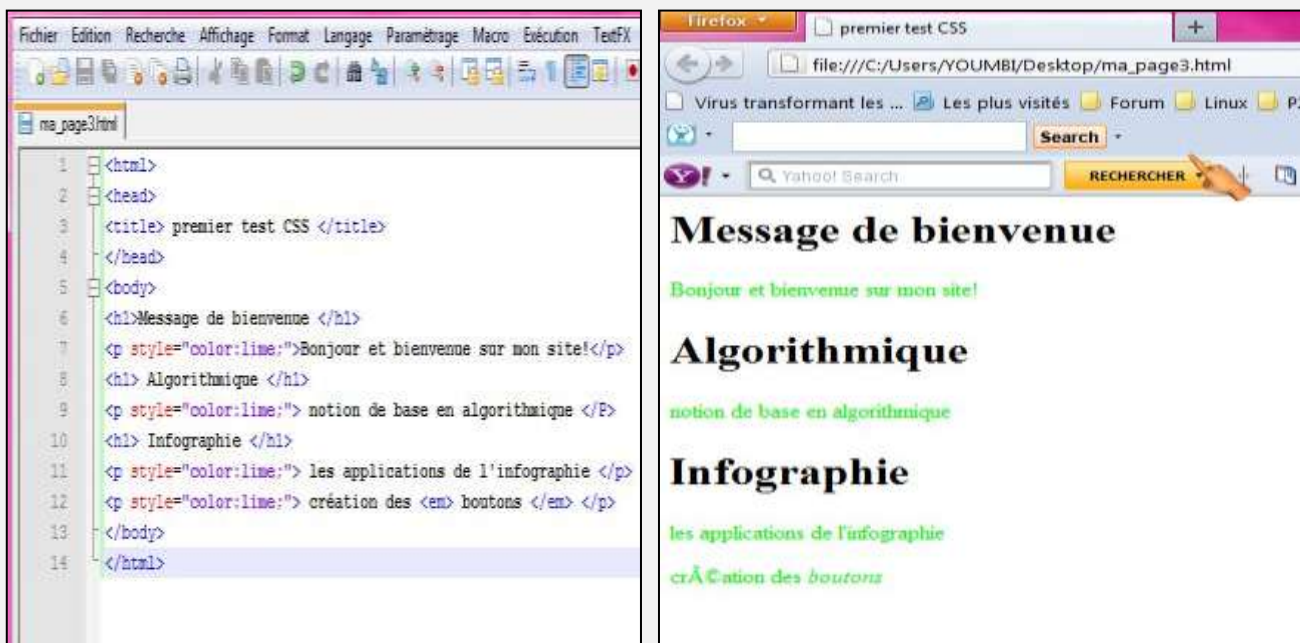
<pre><html> <head> <style> p{color:red;} </style> <title> premier test CSS </title> </head> <body> <h1>Message de bienvenue </h1></pre>	<pre><p>Bonjour et bienvenue sur mon site!</p> <h1> Algorithmique </h1> <p> notion de base en algorithmique </P> <h1> Infographie </h1> <p> les applications de l'infographie </p> <p> création des boutons </p> </body> </html></pre>
---	---

Ouvrir la page web [ma_page2.html](#) pour le tester. On peut constater que tous les textes contenus dans les balises `<p>` sont de couleur rouge.



 tape 3 : Fichier de mise en forme dans les balises du document HTML

  partir du code HTML ci-dessus, cr er une nouvelle page nom ee [ma_page3.html](#), ins rer des codes CSS qui va mettre en vert citron tous les paragraphes de la page web.





1. On peut constater qu'il est plus fastidieux et difficile d'insérer une feuille de style directement dans les balises parce qu'on se retrouve à faire la même opération plusieurs fois. De plus, si on désire changer la mise en forme, il faut modifier chaque balise.
2. Avec un fichier CSS externe, on n'aura besoin d'écrire les instructions de mise en forme qu'une seule fois pour tout le site. On écrit un seul fichier de mise en forme et on l'utilise pour plusieurs pages web en l'insérant à l'entête de toutes les pages à l'aide de l'instruction `< link rel="stylesheet" href="nom_fichier.css" />`
3. On peut insérer des commentaires dans un fichier CSS. Il suffit de placer le contenu dans les signes `/* contenu du commentaire */`



Comment appliquer une feuille de style :

Soit le code html suivant, écrire le Code CSS qui permet de :

1-mettre le titre de la balise `<h1>`

Avec la police « Georgia », couleur Rouge et aligner au centre. Le titre `<h2>` en maron et aligner à Droite.

2-mettre le contenu des balises `<strong>` et `<em>` à une taille plus Grande, souligné et de couleur verte

3-mettre le 1^{er} et 2^{ème} paragraphe en italique, Gras. Mettre le titre de la balise `<h3>` en « Arial Black » et couleur violette. Le mot "débutants" en rose, gras et grande taille.

N.B : il faut noter que chaque fichier CSS créé doit être insérer à l'entête du document HTML à l'aide de la balise `< link rel="stylesheet" href="nom_fichier.css" />`

Étape 1 : application d'un style à une balise

Le lien à insérer est le suivant `< link rel="stylesheet" href="style1.css" />`

```

1 <html>
2 <head>
3 <title> ma page web </title>
4 </head>
5 <body>
6 <h1>BIENVENUE SUR MON SITE</h1>
7 <h2> creation sites web </h2>
8 <h3>Utilisation des feuilles de style</h3>
9 <p> Vous etes au bon endroit ! </p>
10 <p>Ce cours est destine aux debutants qui ne <br/>
11 connaissent rien a la creation de sites web </p>
12 <p>Nous decouvrirons dans ce cours les celebres langages <br/>
13 <strong> HTML5 et CSS3 </strong> utilise pour concevoir des sites web.</p>
14 <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore!</p>
15 </body>
16 </html>

```

```

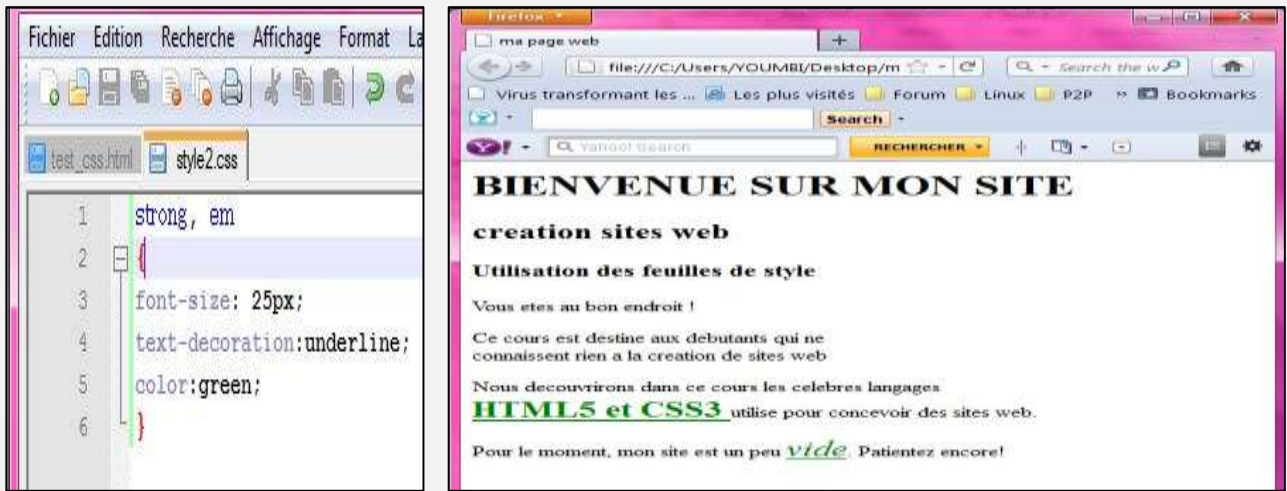
1 h1 {
2 font-family:Georgia;
3 color:red;
4 text-align:center;
5 }
6 h2 {
7 color:maroon;
8 text-align:right;
9 }

```



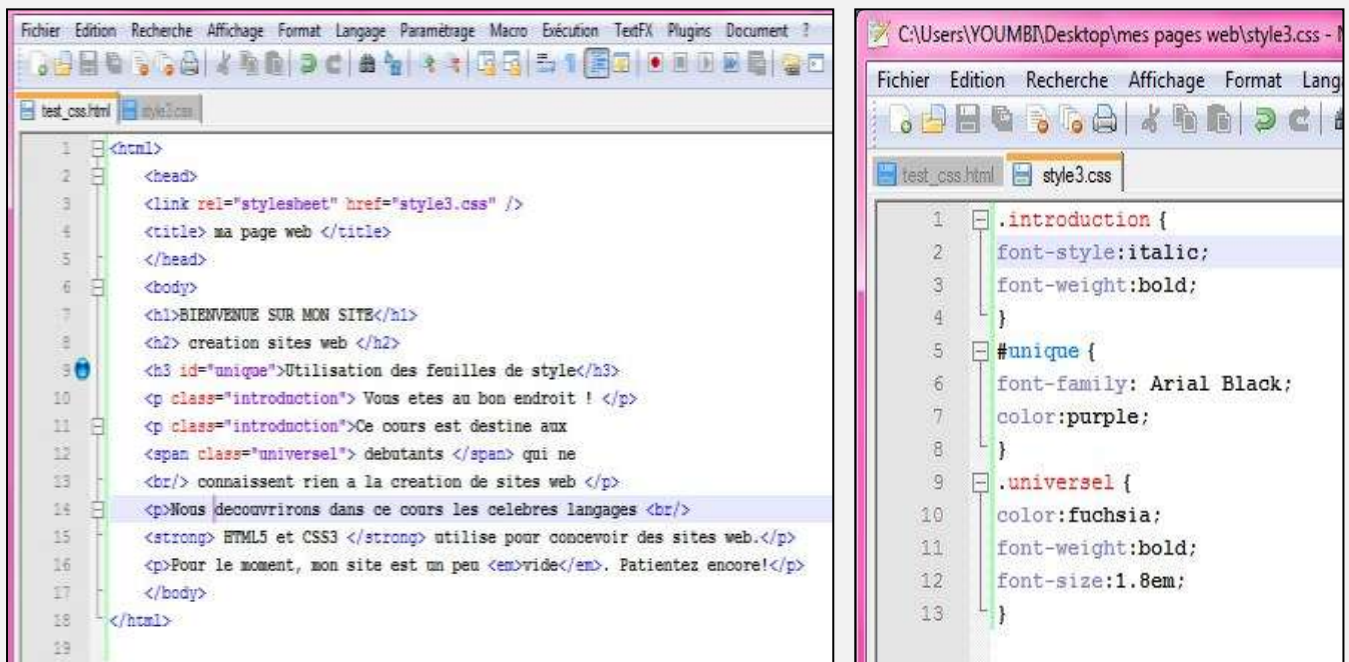
Étape 2 : application d'un style à plusieurs balises

Le lien à insérer est le suivant `< link rel="stylesheet" href="style2.css" />`



Étape 3 : application d'un style à un sélecteur

Il faut utiliser des attributs spéciaux "class" et "id", la balise universelle `<span>`. On a le code html et le fichier CSS ci-dessous. Et le lien : `< link rel="stylesheet" href="style3.css" />`



MODIFICATION D'UNE COULEUR DE FOND

Le lien : `< link rel="stylesheet" href="style4.css" />`



APPLICATION :

Insérer une image à l'arrière plan du document HTML de l'activité à partir d'un fichier css



Compléments

Le **CSS** vous offre un large choix de bordures pour décorer votre page. La propriété permettant de modifier l'apparence des bordures est **border** qui peut utiliser jusqu'à 3 valeurs :

- ⊙ **La largeur** : indique la largeur de la bordure. Mettre une valeur en pixels.
- ⊙ **La couleur** : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un nom de couleur ("black", "red"...), soit une valeur hexadécimale (#FF0000), soit une valeur rgb (rgb(198, 212, 37)).
- ⊙ **Le type de bordure** : la bordure peut être un simple trait, ou des pointillés, ou encore des tirets etc... Voici les différentes valeurs disponibles :

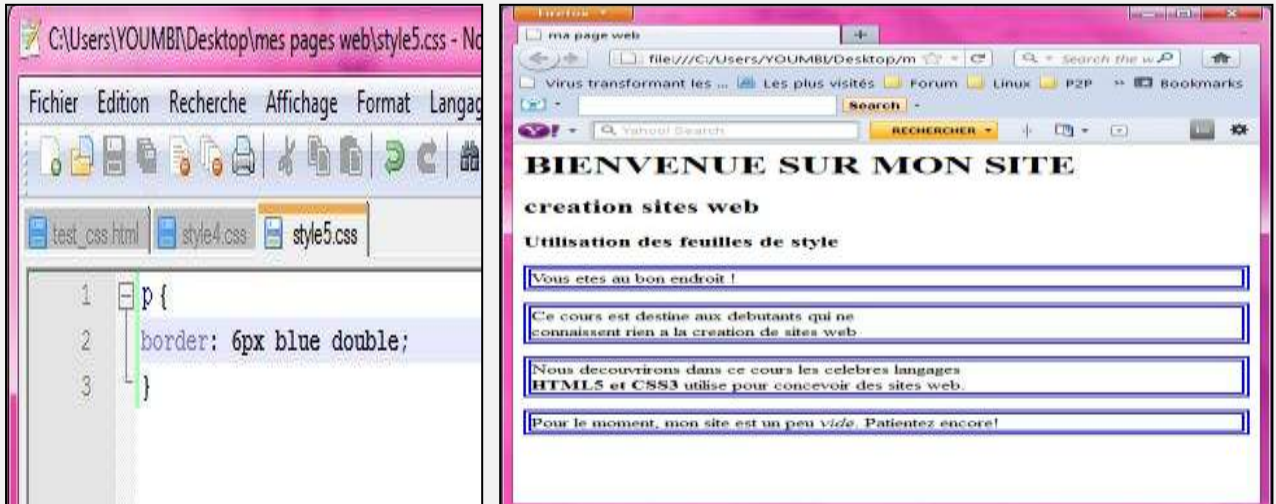
none : pas de bordure (par défaut)	groove : en relief.
solid : un trait simple.	ridge : autre effet relief.
dotted : pointillés.	inset : effet 3D enfoncé.
dashed : tirets.	outset : effet 3D surélevé.
double : bordure double.	

Ainsi, pour avoir par exemple une bordure bleue en tirets de 3 pixels autour des titres de niveau 1, je vais écrire :

```
h1 { border: 3px blue dashed;}
```

EXEMPLE : appliquons une bordure à tous les paragraphes de l'activité précédente .

Le lien est `< link rel="stylesheet" href="style4.css" />`



Résumé

Les feuilles de styles ou encore **CSS (Cascading Style Sheets)** est un langage qui permet de réaliser la **mise en forme** d'une page web et donne au contenu l'apparence qu'on souhaite.

La structure générale d'un fichier CSS est la suivante :

Élément qui peut être une **balise unique**, **plusieurs balises** ou alors un **Sélecteur**.

Les sélecteurs sont des attributs spéciaux qui permettent d'écrire un nom qui sert à identifier la balise sur laquelle sera appliquée la mise en forme.

On distingue 02 attributs spéciaux : **id** et **class**. La différence entre les deux est le fait que le nom donné à l'attribut id ne peut être utilisé qu'une fois dans le fichier HTML alors que celui donné à l'attribut class peut être utilisé autant de fois qu'on le souhaite.

À l'aide du **CSS**, on peut modifier divers aspects d'une page web tels que l'apparence du texte, la couleur ou l'image d'arrière plan, les bordures etc. Pour cela, on utilise des propriétés telles que font-size, font-style, font-family, text-align, text-decoration, background-color, background-image, border, etc.

Le code CSS peut être ajouté à un document HTML de diverses manières :

- ⓐ À partir d'un fichier externe contenant le code CSS et ayant l'extension .css ;
- ⓑ Dans l'entête du document HTML à l'aide de la balise `<style>.....</style>`
- ⓒ Dans les balises du document HTML à l'aide de l'attribut style.

La meilleure méthode est un fichier externe car elle facilite les modifications et elle offre la possibilité de pouvoir appliquer une même mise en forme à plusieurs pages web.

Élément{
propriété: valeur;
propriété: valeur;
propriété: valeur;
}



EXERCICE 1:

- 1) Définir et donner le rôle d'une feuille de style
- 2) Il existe plusieurs façons d'appliquer une mise en forme à un document HTML. Citer les en donnant des exemples
- 3) Quelle propriété permet de modifier la couleur de fond d'une page web ? donner un exemple d'utilisation.
- 4) Quelle propriété permet d'appliquer une bordure ? donner un exemple d'utilisation.
- 5) Où peut-on placer du code CSS ? donner des exemples à chaque fois.

EXERCICE 2:

Créer deux pages web : la première "ma_page.html" et la seconde "mon_lycee.html". Créer un fichier de mise en forme externe appelé mon_style.css.

- 1) Quelle balise permet de lier un fichier HTML à un fichier CSS externe ?
- 2) Appliquer cette mise en forme aux deux fichiers HTML.
- 3) Appliquer une couleur d'arrière plan à "ma_page.html" et une image d'arrière plan au second document HTML.

EXERCICE 3:

```

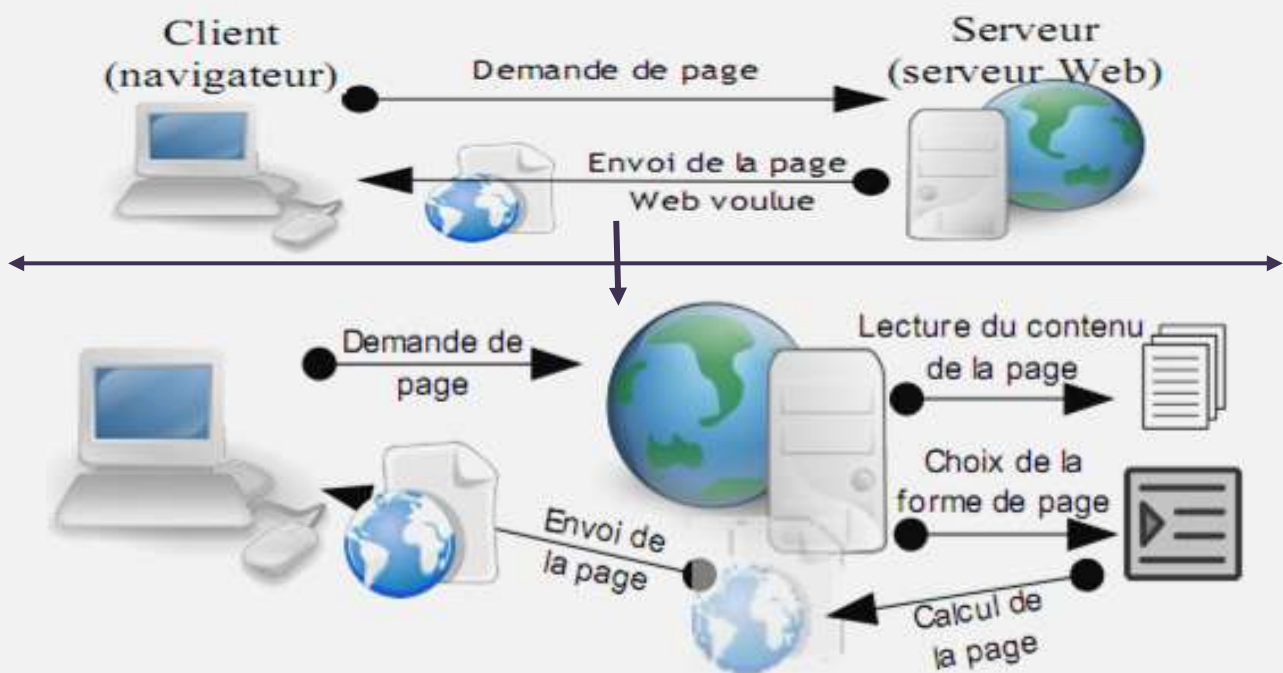
1 <html>
2 <head>
3 <title> ma page web </title>
4 </head>
5 <body>
6 <h1> BONJOUR ET BIENVENUE SUR CE SITE </h1>
7 <h2> Découverte des calques </h2>
8 <p> On pourrait définir un calque comme étant <br/>
9 un élément graphique qui est géré par Photoshop. <p>
10 <p> chaque élément dans une zone de travail possède son <br />
11 propre calque que l'on retrouve dans la fenêtre de calques. </p>
12 <h2> Créer un calque </h2>
13 <p> On a la possibilité dans un premier temps de cliquer sur l'icône <br />
14 Nouveau calque Image utilisateur qui se situe en bas de la fenêtre des <br/>
15 calques. C'est la façon la plus rapide de créer un calque. </p>
16 </body>
17 </html>
  
```

Écrire le code CSS qui permet de :

- mettre en bleu, italique les deux premiers paragraphes ;
- utiliser la police "Verdana" pour tous les paragraphes
- mettre les titres de niveau 1 et niveau 2 au centre, souligné et en rouge
- mettre l'expression "zone de travail" avec une grande taille de police, en gras et de couleur rose.
- Appliquer une bordure en pointillés aux deux derniers paragraphes.
- Appliquer une couleur d'arrière plan

Chapitre 14

INTRODUCTION AUX SITES WEB DYNAMIQUES



SOMMAIRE



- I- LES LIMITES DU SITE WEB STATIQUE
- II- DU SITE STATIQUE AU SITE DYNAMIQUE
- III- FONCTIONNEMENT D'UN SITE WEB DYNAMIQUE



Pré-requis

- Les éléments de base pour la création d'un site
- Connaître ce que c'est qu'un site statique
- Le fonctionnement d'un site statique



Objectifs

- Connaître les limites d'un site web statique
- la différence entre un site statique et un site dynamique
- le fonctionnement d'un site dynamique

INTRODUCTION

Une **page Web dynamique** est une page Web générée à la demande, par opposition à une **page Web statique**. Le contenu d'une page Web dynamique peut donc varier en fonction d'informations (heure, nom de l'utilisateur, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation. À l'inverse, le contenu d'une page Web statique est *a priori* identique à chaque consultation.

Lors de la consultation d'une **page Web statique**, un **serveur HTTP renvoie le contenu du fichier** où la page est enregistrée. Lors de la consultation d'une **page Web dynamique**, un **serveur HTTP transmet la requête** au logiciel correspondant à la requête, et le logiciel se charge de générer et d'envoyer le contenu de la page. La **programmation Web** est le domaine de **l'ingénierie informatique** dédié au développement de tels programmes.

I. LES LIMITES DU SITE WEB STATIQUE

Dans un **Site web statique**, *les pages du site ne sont pas modifiables par des utilisateurs*. Le site est donc rempli et mis à jour par l'administrateur qui le fait depuis son poste de travail. Une fois le site mis à jour sur l'ordinateur de l'administrateur, celui-ci devra être envoyé sur le site via FTP. Le site est dit statique car les pages HTML qui le compose sont toujours identiques entre deux visites sans mise à jour. Le serveur donc n'a pas besoin de éléments de scripting (Php, Perl, Python) et de base de données (MySQL, SQLite) pour faire fonctionner le site, puisqu'il sert directement des pages HTML.

On utilisera un site web statique pour une situation bien particulière. On utilisera ce fonctionnement **pour un site web nécessitant peu de maintenance, peu de mise à jour et contenant peu de pages** (même si ce dernier problème peut être résolu en faisant un peu de programmation).

En effet, comme la mise à jour d'un site Internet statique peut être fastidieuse, on utilisera ce type de site uniquement si les mises à jour sont exceptionnelles. Car, à chaque

mise à jour il faudra modifier la page HTML du site et la mettre en ligne en effectuant une copie par FTP.

Mais il faut aussi reconnaître au site statique des avantages dans plusieurs domaines du site internet :

- ⊗ le site internet est mis à jour en local sur la machine de l'administrateur : il n'y a donc pas de **mauvaise** surprise une fois que le site est en ligne. Le site internet ne fait pas appel aux technologies en perpétuelles évolutions qui permettent la mise en place de sites dynamique (**Php, Ruby, Python, Perl, Java, ASP, etc.**) : on gagne donc en **sécurité** et en **veille technologique**.
- ⊗ le site internet statique consomme peu de ressource serveur : le site n'utilisant aucune technologie compliquée (au hasard : **Php + MySql + Apache**), les coûts d'entretien et de maintenance en activité sont très inférieurs à ceux d'un site dynamique
- ⊗ le site internet statique se sauvegarde plus facilement : ceux qui ont déjà manipulé les bases de données MySQL utilisées pour la création de sites dynamiques savent que c'est une galère à sauvegarder et à restaurer des données importantes. Le fait de disposer directement des pages HTML du site facilite la sauvegarde (un simple copier / coller sur une clé USB et c'est bon !)

On utilisera plutôt un site dynamique si on souhaite créer une interaction avec ses utilisateurs. Le site dynamique permet de se connecter en ligne sur son site pour réaliser sa mise à jour en direct. Une fois la mise à jour du site dynamique effectuée, le résultat apparaît directement aux utilisateurs.

On privilégiera donc les sites web dynamiques pour les sites permettant aux visiteurs de laisser des commentaires (blogs) ou de converser avec d'autres utilisateurs (forums). De même, si le site doit être mis à jour très fréquemment (plusieurs fois par semaine) on pourra choisir de créer un site dynamique même si aucune interaction n'est prévue avec les visiteurs.

Par exemple pour réaliser un site vitrine présentant les 10 produits vendus par une entreprise on pourra choisir :

- **Un site statique** si les produits ne sont modifiés qu'une ou deux fois par mois et qu'un ou deux nouveaux produits sont ajoutés au catalogue chaque année.
- **Un site dynamique** si on souhaite que les visiteurs ajoutent des commentaires sur les fiches des produits et/ou qu'il faut modifier très souvent (ajout, suppression, modification) les fiches des produits.

II. DU SITE WEB STATIQUE AU SITE WEB DYNAMIQUE

Dans un **site web dynamique**, les pages peuvent être modifiables par les visiteurs. De plus, entre deux visites sur un même site, le contenu de la page peut être différent sans action de l'administrateur du site Internet. Les grandes applications de ce type de site sont : les **forums**, les **Wiki** (Wikipedia étant le plus grand représentant du genre) et tous les sites communautaires (**Facebook, Twitter, his, etc.**). Contrairement au site web statique, le serveur qui fait fonctionner le site web dynamique utilise une technologie de scripting (comme **Php, Ruby, Python ou Perl**) ainsi qu'une base de données comme **Mysql**.

Pour passer du site web statique au site web dynamique, plusieurs éléments entre en jeu parmi lesquels les plus majeurs sont : le **JavaScript** et les bases de données.

1. **JavaScript**

JavaScript n'est pas directement une partie qui constitue HTML, mais un langage de programmation distinct. Ce langage a pourtant été créé spécialement pour mettre à la disposition des auteurs un outil grâce auquel les pages WWW se laissent optimiser.

Les programmes en JavaScript sont notés directement dans le fichier HTML ou dans des fichiers séparés. Ils sont interprétés en temps réel par le navigateur Web. Les navigateurs Web modernes possèdent à cet effet le logiciel d'interprétation correspondant.

Dans un langage de programmation comme JavaScript il y a beaucoup d'éléments prêtant à confusion pour les débutants; ce sont les caractères spéciaux, les variables, les commandes si-alors, les boucles, les fonctions, les méthodes, les paramètres, les objets, les propriétés et plus encore. Pour pouvoir manier facilement ces éléments vous devez apprendre à vous représenter ce qui se passe dans l'ordinateur quand le code du programme est exécuté. C'est un apprentissage pénible que des débutants ne maîtriseront que par des exercices répétés. Pour cela, JavaScript est très bien approprié parce qu'il est un langage de programmation relativement simple, dans lequel manquent beaucoup de tâches d'un "grand" langage de programmation - comme par exemple la gestion de la mémoire ou le maniement de fichiers. En outre, JavaScript est employé dans un environnement déterminé, à savoir la page WWW affichée ou à afficher. Il est indispensable pour la création d'un site web dynamique.

2. **Base de données**

Une **Base de données** est un ensemble d'informations structurées mémorisées sur un support permanent.

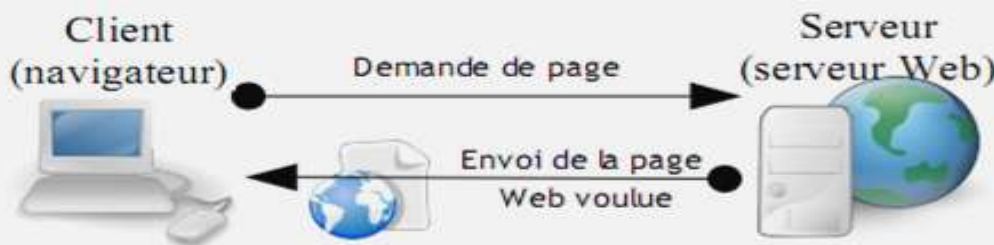
On peut remarquer qu'une organisation consistant en un (ou plusieurs) fichier (s) stockés sur mémoire secondaire est conforme à cette définition. Un ensemble de fichiers ne présentant qu'une complexité assez faible, il n'y aurait pas là matière à longue dissertation. Malheureusement l'utilisation directe de fichiers soulève de très gros

problèmes tels que la lourdeur d'accès aux données, le manque de sécurité, absence du contrôle de concurrence.

D'où le recours à un logiciel chargé de gérer les fichiers constituant une base de données, de prendre en charge les fonctionnalités de protection et de sécurité et de fournir les différents types d'interface nécessaires à l'accès aux données. Ce logiciel est le Système de Gestion de Base de Données (SGBD). Un Système de Gestion de Bases de Données est un logiciel de haut niveau qui permet de manipuler les informations stockées dans une base de données.

III. FONCTIONNEMENT D'UN SITE WEB DYNAMIQUE

Un **site Web statique** est un site où chacune des pages est créée en HTML. Un ordinateur qui se connecte au serveur, demande une page. Celle-ci lui est directement servie (elle est stockée toute prête sur le serveur) :



Par opposition, un **site Web dynamique** est un site Web dont les pages sont générées dynamiquement à la demande. Le contenu est obtenu (par exemple) en combinant l'utilisation d'un langage de scripts ou de programmation et une base de données. Il s'agit souvent de PHP pour le langage et MySQL pour la base de données.

Dans les sites dynamiques le contenu (articles) est séparé de l'habillage (modèles ou squelette). Cette séparation contenu/présentation/logique est le credo des développements actuels. Les avantages sont donc non négligeables et les possibilités de dynamisation évoluent de jour en jour.



Le site web dynamique a aussi des avantages :

- de nombreux scripts gratuits existent déjà et permettent de réaliser tous les sites qu'on souhaite. Ainsi en téléchargeant le script (ou CMS : Content Management System) qui va bien, il sera très simple de créer un forum, un blog ou tout autre site.
- La mise à jour est très simple : une fois le script dynamique en place, on met à jour le site en ligne dans la partie « administration » du site. On peut donc mettre à jour le site de n'importe quel ordinateur et même depuis certains téléphones mobiles (avec accès Internet naturellement)
- Avec un site dynamique il est possible de réaliser une grande interaction avec les visiteurs : les visiteurs peuvent donc rester beaucoup plus longtemps sur vos pages si les fonctionnalités sont intéressantes.

CONCLUSION

Il n'y a pas une solution meilleure que l'autre. Que le site soit statique ou dynamique, c'est le cahier des charges qui va faire la différence. Il faut donc être extrêmement attentif aux fonctionnalités à mettre en place (veuillez aussi considérer la problématique des commentaires laissés par les visiteurs).

Mais, pour résumer très simplement, voici les informations qu'on peut retenir :

- Le site ne sera pas mis à jour souvent : **site statique**
- Il s'agit d'un site vitrine et votre budget est limité : **site statique**
- Vous souhaitez un forum pour que vos visiteurs puissent échanger sur votre site : **site dynamique**
- Vous souhaitez récolter des informations sur vos visiteurs : **site dynamique**.

MODULE

4

PROJET

RÉALISATION
D'UN
SITE WEB
STATIQUE

INTRODUCTION

On parle de **projets** dans la vie de tous les jours à l'instar de nos projets de vacances, projets de carrière, projets d'avoir des enfants... Cette notion de **projet** nous vient du latin « **projectum de projicere** », qui signifie littéralement « jeter quelque chose vers l'avant ».

Un **projet** est une chose ou un ensemble de choses que l'on se propose de faire en temps donné, mettant en œuvre des ressources humaines et matérielles faisant l'objet d'une budgétisation, et aboutissant à un ensemble de livrables. Dans notre contexte cette chose est un site web.

La mise en œuvre d'un projet passe par plusieurs étapes à savoir l'analyse, la conception et la réalisation. Dans chaque étape, un ensemble de tâches doivent être réalisés.

ANALYSE

L'analyse d'un projet est une phase primordiale dans la conduite d'un projet. Sans toutefois faire un cours sur la gestion des projets ou sur la conduite d'un projet, cette étape permet en quelque sorte de formaliser les besoins (fonctionnels et non fonctionnels) de votre projet, de répondre à la question du Quoi ?

Dans le cadre de l'analyse d'un site web, il faut définir les objectifs à atteindre ; c'est-à-dire les fonctionnalités que doit avoir le site web.

Tout d'abord le site web à construire est un site web statique. En d'autres termes, c'est un site web dont les pages sont figées, les modifications ne sont faites que par l'administrateur du site. C'est un travail récapitulatif du programme d'informatique. Les attentes sont les suivantes :

- ⊕ Le projet consiste en la mise en œuvre d'un site web intégrant différents modules, chaque module étant le site web d'une discipline de la classe de Première.
- ⊕ Chaque discipline comprend des rubriques (cours, exercices, travaux pratiques) devant être détaillées dans le site. ces rubriques peuvent également avoir des sous rubriques
- ⊕ Le site doit avoir une page d'accueil comportant le logo de l'établissement, une bannière, un menu, une zone de publicité, ...

- ⊕ On doit pouvoir accéder à une discipline à partir de la page d'accueil principale
- ⊕ La navigation doit être facile
- ⊕ On doit pouvoir passer d'une page à une autre via un lien hypertexte
- ⊕ Toutes les pages doivent avoir la même mise en forme c'est-à-dire les mêmes couleurs, polices, taille des polices, la façon de présenter les titres, les liens, les liens visités, ...
- ⊕ L'établissement doit être globalement présenté à l'accueil
- ⊕ L'interface doit avoir des petites animations attrayantes

CONCEPTION

Cette phase de la conduite d'un projet prend en entrée la spécification des besoins qui découle de la phase d'analyse, et en sortie génère un document qui décrit comment le projet sera réalisé. Cette étape permet de répondre à la question du **Comment?**

I. ORGANISATION DU CONTENU

Un site web présente un ensemble d'informations sur des pages web. La page web étant l'élément de base du site, elle doit présenter les informations les plus détaillées possibles. Par exemple, les pages web du site d'une bibliothèque vont présenter des livres, les pages web d'un concessionnaire de voitures vont présenter des voitures. Chacune de ces pages peut être regroupée en rubrique ou en sous-rubrique (pour le site de la bibliothèque on pourra regrouper les livres par genre, puis par langue). Toutes ces pages sont interconnectées les unes aux autres par des liens.

Pour notre site, les pages web vont présenter les disciplines d'une classe particulière. Une de ces pages sera la page d'accueil ; celle à partir de laquelle toutes les autres pages seront accessibles.

Les contenus sont organisés comme suit :

- ✍ Une page d'accueil qui contiendra :
 - ⊕ Un logo qui sera celui de votre établissement. Il sera conçu dans un logiciel d'infographie grâce à toutes les connaissances reçues dans le domaine de l'infographie
 - ⊕ Une bannière qui présente plusieurs éléments de votre choix entre autre des images de votre établissement, un drapeau flottant (Le drapeau flottant est le résultat d'une animation), etc ;

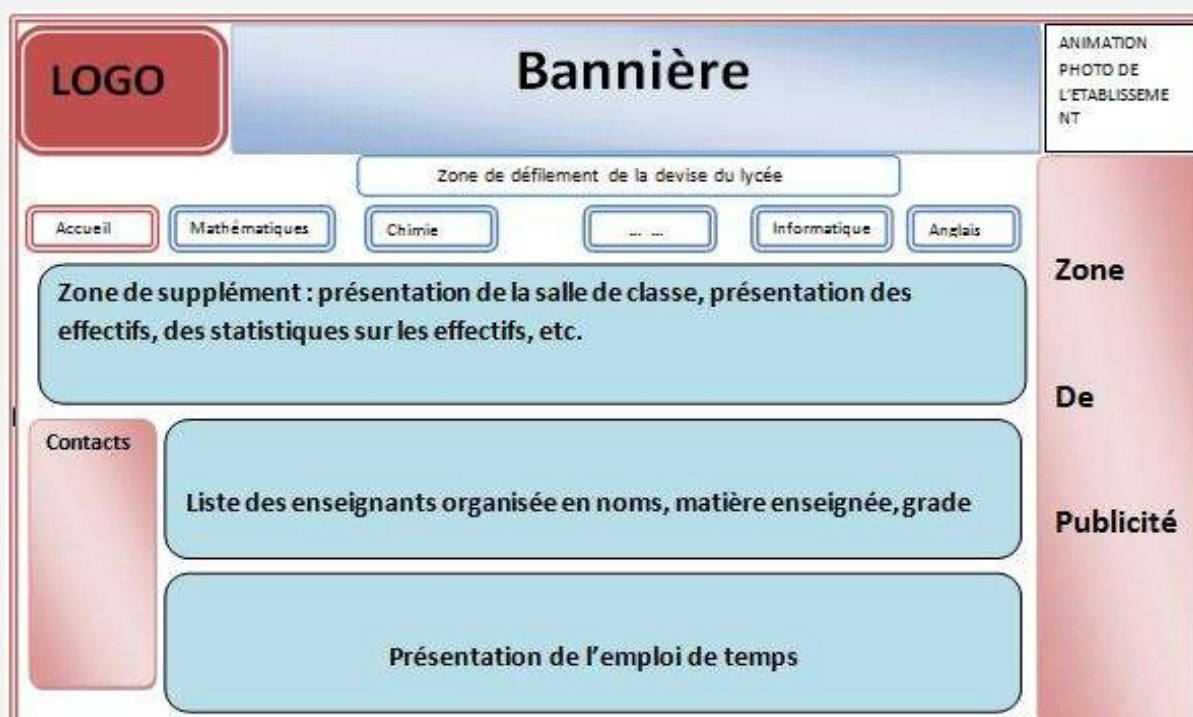
- ⊕ Une zone où défile la devise de l'établissement scolaire
 - ⊕ Une zone ou barre de menu (un menu peut être un bouton, un texte, une image, etc). Chaque menu sera le nom d'une matière et c'est à partir d'un clic sur un menu que l'on accèdera à la page décrivant la matière concernée. Il faudra réserver un menu **contact**. En cliquant sur ce menu, on obtiendra la liste des contacts des personnes ayant réalisé le site
 - ⊕ Un espace publicitaire qui présente quelques publicités sur l'établissement
 - ⊕ Une zone de sous menu qui correspond aux sous rubriques d'un menu
 - ⊕ Une zone de contenu qui sera sectionnée en 3 comprenant l'emploi de temps, la liste des enseignants (noms, grade, matière enseignée), un supplément d'information (présentant la classe, l'examen en vu, les statistiques sur les élèves)
- ✎ Les différentes pages de cours. Chacune de ces pages aura le même logo, la même bannière, le même espace publicité. Les éléments qui varient sont la zone de contenu, les sous menus qui seront réorganisés en fonction de la discipline

2. CONCEPTION DE LA MAQUETTE DU SITE

La maquette du site repose sur le gabarit qui est une disposition standard que toutes les pages web du site devront avoir. Généralement les sites web ont une disposition standard. Il s'agit :

- ⊕ **D'un bloc d'en-tête** : qui comporte généralement le logo de l'entreprise, le nom du site, un slogan, une bannière ;
- ⊕ **D'un bloc de menu** : qui contient les liens vers les rubriques ou les sous rubriques ;
- ⊕ **D'un bloc de contenu** qui contient les contenus des pages.

Un exemple de maquette de la page d'accueil est la suivante :



NB : Tous les éléments spécifiés dans la phase d'analyse doivent apparaître dans le site web.

Un exemple de maquette pour une discipline (Mathématiques) est la suivante :



3. CONCEPTION DES CODES HTML

Dans cette rubrique, il faut donner tous les codes html qui vont permettre de réaliser le site. Quelques directives facilitant la réalisation de cette tâche sont présentées ci-dessous.

Page d'accueil

A l'aide de vos connaissances en infographie,

- ② Concevoir le logo de votre site
- ② Construire une animation pour votre établissement
- ② Construire une bannière pour votre site web
- ② Construire les boutons (deux versions) pour chaque matière.

A l'aide de vos connaissances sur la programmation html

- ② Écrire une feuille de style pour gérer la mise en forme des pages html
- ② Dans le code principal (page d'accueil nommée **index.html**)
 - ▶ donner une couleur d'arrière plan au choix
 - ▶ l'astuce pour obtenir l'interface précédente est la suivante : insérer un tableau (six lignes, trois colonnes) Faites des fusions de cellules pour obtenir une image semblable à la suivante :

Logo	Bannière	animation
Zone de défilement de la devise ; Fusion de deux colonnes de la deuxième ligne		Zone de publicité Fusion des cinq dernières lignes de la dernière colonne
Insérer les menus ; fusion des deux colonnes de la troisième ligne		
Suppléments ; fusion des deux colonnes de la quatrième ligne		
Zone de contact ; fusion des deux dernières lignes de la première colonne	Liste des enseignants	
	Emploi de temps	

- ④ insérer les éléments nécessaires dans ce code index.html
- ④ pour plus de lisibilité, les zones de suppléments, liste enseignant et emploi de temps doivent avoir des barres de défilement. Pour cela insérer les frames et imposer qu'ils aient des barres de défilement.
- ④ Pour chaque menu, faire un lien vers la page adéquate (qui peut être un site).

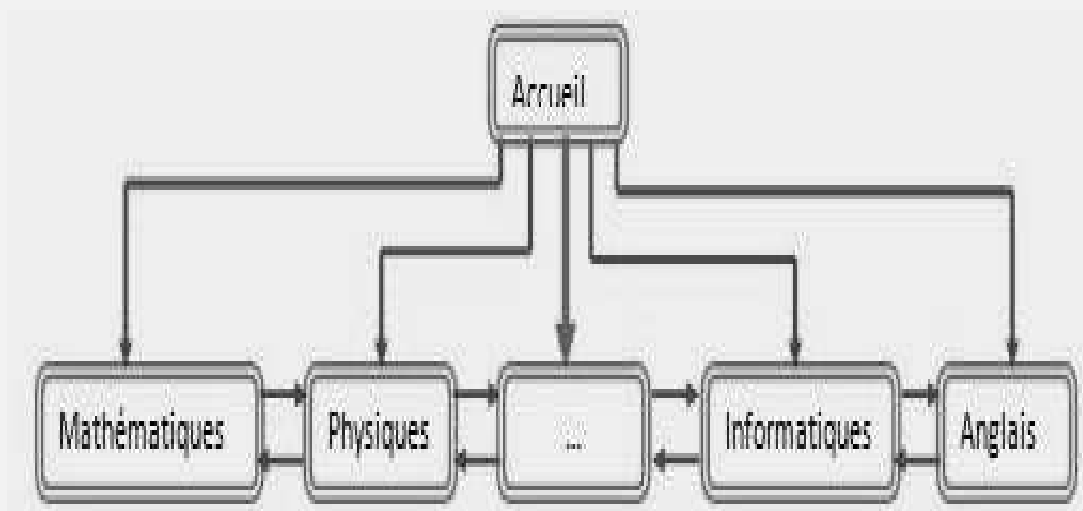
Autre Page

Chaque matière est représentée dans une ou plusieurs pages. Les sous rubriques peuvent être des liens vers un emplacement dans la même page, ou vers une autre page du site.

S'inspirer de ce qui a été fait dans la page d'accueil pour construire ces différentes pages.

4. PLAN DE NAVIGATION DU SITE

Le plan de navigation est donné comme suit :



Les boutons symbolisent les pages web et les flèches représentent les liens de navigation entre les pages.

RÉALISATION

Cette étape consiste en la mise en œuvre du résultat de la conception. Le travail sera d'écrire les codes html comme détaillés dans la phase de conception.

PUBLICATION D'UN SITE WEB

Après avoir réalisé un site web sur un ordinateur, celui-ci n'est visible qu'en local sur cette machine. Si on désire que le site soit visible par d'autres personnes via Internet, il faut le publier. Pour ce fait, la publication ou hébergement passe par trois étapes : la réservation d'un nom de domaine, le choix de l'hébergeur et la publication proprement dite.

1. Réservation du nom de domaine

Il s'agit en fait d'une adresse sur le Web constituée de deux parties.

On a par exemple **ens-dite.com**

En **cyan (vert clair)**, le **nom de domaine** proprement dit qui peut être choisi librement tant que personne d'autre ne l'a déjà réservé. Il peut contenir des lettres et des chiffres, mais pas de symboles particuliers (comme le **ç**, le **é**, le **è**, les **espaces**, etc).

En **rose**, l'**extension** (aussi appelée **tld**). Il existe grosso modo une extension par pays (**.fr** pour la France, **.be** pour la Belgique, **.ca** pour le Canada). Toutefois, il y a aussi des extensions utilisées au niveau international comme **.com**, **.net**, **.org**.

Pour réserver un nom de domaine, il faut passer par un organisme chargé de la gestion des noms de domaines au niveau national ou international. Cette réservation est payante et le montant varie en fonction de l'extension (**tld**) choisie.

2. Le choix d'un hébergeur

Sur Internet, tous les sites web sont stockés sur des ordinateurs particuliers appelés "**Serveurs**". Ce sont des ordinateurs généralement très puissants qui restent tout le temps allumés. Ils contiennent les pages des sites web et les délivrent aux internautes qui les demandent, à toute heure du jour et de la nuit.

L'**hébergeur** est une entreprise qui se charge de gérer les serveurs. Elle s'assure du bon fonctionnement des serveurs 24h/24 7j/7. En effet, si l'un d'eux tombe en panne, tous les sites présents sur la machine deviennent inaccessibles rendant les visiteurs du site mécontents. Les hébergeurs sont très nombreux et sont de tous types, à tous les prix ; on en a aussi des gratuits. Ils offrent un espace de stockage (permettant de stocker les dossiers du site) et un trafic (Le trafic, c'est la quantité de données envoyées par mois aux visiteurs d'un site web).

Quelques hébergeurs : **MavenHosting**, **PlanetHoster**, **I&I** etc. Pour avoir d'autres hébergeurs, il suffit de lancer une recherche sur ce thème dans la fenêtre d'un moteur de recherche sur internet.

3. L'hébergement proprement dit

Pour publier un site web :

- ⊕ Il faut se connecter au site web de l'hébergeur choisi ; fournir les informations requises telles que le nom de domaine, le type d'hébergement etc.

Transférer les fichiers et dossiers du site vers le serveur de l'hébergeur. Pour le faire, il faut se servir des protocoles **FTP** et **HTTP**. **FTP** signifie **File Transfer Protocol**

c'est le moyen que l'on utilise pour envoyer les fichiers vers le serveur de l'hébergeur. Il existe de nombreux logiciels (gratuits ou payants) permettant d'utiliser le **FTP** pour transférer les fichiers sur Internet. Ceux qui sont gratuits sont téléchargeables sur le net. Le **HTTP** qui signifie **Hyper Text Transfer Protocol**, va permettre aux visiteurs du site de visionner les fichiers avec un navigateur.

⊕ Quel que soit l'hébergeur choisi, cela fonctionne toujours de la même manière. Il fournit 3 informations qui sont indispensables pour que **le serveur FTP** puisse se connecter au serveur web (serveur de l'hébergeur) :

- Ⓢ **L'IP** : c'est "**l'adresse**" du **serveur web**. Le plus souvent, on vous donnera quelque chose du genre **ftp.mon-site.com**, mais il peut aussi s'agir d'une suite de nombres comme **122.65.203.27**
- Ⓢ **Le login** : c'est **l'identifiant** fourni par vous lors de l'hébergement du site. ça peut être un pseudo, le nom de votre site.
- Ⓢ **Le mot de passe** : soit on vous a demandé un mot de passe, soit on vous en a attribué un d'office.

THÈMES DES PROJETS

Thème 1 : SITE WEB DE LA DISCIPLINE INFORMATIQUE

Thème 2 : SITE WEB DE LA DISCIPLINE PHYSIQUE

Thème 3 : SITE WEB DE LA DISCIPLINE MATHÉMATIQUES

Thème 4 : SITE WEB DE LA DISCIPLINE CHIMIE

Thème 5 : SITE WEB DE LA DISCIPLINE FRANÇAIS

Thème 6 : SITE WEB DE LA DISCIPLINE ANGLAIS

Thème 7 : SITE WEB DE LA DISCIPLINE HISTOIRE

Thème 8 : SITE WEB DE LA DISCIPLINE GÉOGRAPHIE

Thème 9 : SITE WEB DE LA DISCIPLINE SCIENCES NATURELLES

Thème 10 : RÉALISATION DE LA PAGE D'ACCUEIL PRINCIPALE (ce thème vise la mise en œuvre de page d'accueil qui permettra au visiteur de se rendre vers le site de n'importe quelle discipline)

INDEX

A

action, **69**, Voir opération, acte, geste, démarche
affectation, **45**, Voir transfert
ajuster, **20**, Voir affiner
algorithme, **45**
algorithmique, **45**
alternative, **69**, Voir option, choix
animation, **20**
architecture
 architecte, **9**

B

balise, **89**, Voir marque, trace, indication, etc.
boucle, **75**, Voir noeud, anneau, maillon, etc.
bouton, **20**

C

calque, **17**, Voir feuille de dessin
cartographie
 carte, **9**
choix, **69**, Voir option, préférence, désignation,
 indécision, etc.
condition, **69**, Voir clause, exigence, modalité

D

déclaration, **45**
décrémentation, **45**
déverrouiller, **20**, Voir débloquent

E

éditeur de texte, **89**
enchaînement, **65**
en-tête, **45**
exécution, **69**, Voir application
extraction, **20**

F

filtre, **20**, Voir tamis
fonction, **45**

G

graphisme, **9**

I

identificateur, **45**
incrémentation, **45**
infographe, **9**, Voir infographiste
infographie, **8**
informatique, **9**
instrcution, **45**
instruction, **69**, Voir ordre, commande
itération, **69**, Voir répétition, rabâchage

L

langage, **89**, Voir expression, vocabulaire, jargon, etc.
logiciel, **10**, Voir programme
logos, **20**

N

navigateur, **89**

O

organigramme, **75**

P

palette, **20**
procédure, **45**

R

récurtivité, **65**

S

séparateur, **45**
structure, **65**, **69**, Voir composition, agencement,
 combinaison, disposition, arrangement, etc.
style, **20**

V

variable, **45**
visualisations
 visualiser, **9**



www.dite-ens.com



Copyright © 2012

Promotion Arc-en-Ciel
 PROMOTION ARC-EN-CIEL